A Reproduced Copy

OF

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

DRF-9612

TR-796-9-594
12 AUGUST 1969

LAUNCH VEHICLE TRAJECTORY OPTIMIZATION COMPUTER PROGRAM - PHASE IV

FINAL REPORT

PREPARED FOR:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER

UNDER CONTRACT NAS8-30509

NORTHROP CORPORATION
ELECTRO-MECHANICAL DIVISION
P. O. BOX 1484
HUNTSVILLE, ALABAMA 35807
TELEPHONE (205)837-0580

TR-796-9-594

LAUNCH VEHICLE TRAJECTORY OPTIMIZATION COMPUTER PROGRAM – PHASE IV

Final Technical Report

12 August 1969

By

C. B. Hose, Jr.
T. D. Neeley
D. R. Kilpatrick

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER

Under Contract NAS8–30509

REVIEWED AND APPROVED BY:

Douglas Raney, General Supervisor,
Flight Mechanics Branch

W. A. Klabunde, Director
Astrodynamics Section

NORTHROP CORPORATION
HUNTSVILLE, ALABAMA

FOREWORD

The Automatic Performance Trade-off Computer program (APT) described herein was designed, coded, and debugged while Northrop was under a fixed price contract (NAS8-30509) to MSFC. The Contracting Officer's Representative throughout the study has been Mr. Bill Goldsby of the Flight Mechanics Branch, Program Development Directorate.

The authors would like to acknowledge Mr. Goldsby and the members of his staff for their helpful support in providing performance data and for their technical contributions to the design of the program.

TABLE OF CONTENTS

ABSTRACT

This report describes a computer program which retrieves stored data and determines variations in launch vehicle performance as a function of mission and vehicle parameters. The data stored by the program were generated by sophisticated launch vehicle optimization programs. This program was developed as a tool for mission planning permitting an instant response capability for preliminary problem analysis.

Cataloging techniques developed for various missions are described. These techniques determine the points which must be stored for a particular mission. Large quantities of data for various launch vehicles have been stored in the program and by using interpolation and extrapolation techniques new data may be generated from this stored information.

Section I

INTRODUCTION

1.1 BACKGROUND

The need for quick response by mission planners and the large amounts of optimum performance data that have been generated for a number of launch vehicles of interest has resulted in the NASA request (RFQ DCN 1-7-21-0002) for a quick retrieval system for surveying optimum performance variations due to mission and vehicle variations. Described in this final technical report is the Automatic Performance Trade-off (APT) computer program which is Northrop's response (under NAS8-30509) to the NASA need. To satisfy the needs of mission planners such a system should have the following characteristics. First, it should efficiently catalog and store information from optimum launch vehicle trajectories in large data banks. Secondly, it should be able to quickly retrieve these data and by using numerical techniques expand their range of usefulness. Finally, it should display these data in several formats each of which have been designed to maximize the immediate usefulness of the data to a particular mission planning function. The Northrop Automatic Performance Trade-off program completely satisfies these requirements.

The benefits to the mission planners from such a system are many fold:

- The efficient utilization of the APT program will tend to eliminate wasteful duplication in the generation of performance data in connection with mission planning activities. Thus, the cataloging and storage system formally gives a permanent "home" to any generated data. This will eliminate the necessity of running most performance variation prediction cases because sufficiently accurate data for all reasonable data points can be found by interpolation and extrapolation of existing numbers in the storage bank.
- Because of the generality of the numerical techniques, data trends, and program logic; newly evolving launch vehicles can be conveniently added to the storage banks by running a minimum number of selected optimum trajectories for data points.
- The program can be asked to display the extent and limitations of its data with respect to any or all of the launch vehicles and a figure of merit is available for all extrapolated payload numbers.

- In addition, if, (1) data points exist for the vehicles of interest, and (2) all vehicle and mission variations are reasonable; then, all mission planners associated with a given study can be immediately provided with new performance data - in a format streamlined for their immediate needs - whenever study ground rules change. It is believed that this quick response will be of major importance in many studies and allow planners to concentrate on more subtle aspects of the study.

1.2 OBJECTIVES

The scope of work as stated in the contract (NAS8-30509) was to devise a computerized technique by which stored data generated from a sophisticated launch vehicle trajectory optimization program can be speedily transformed into meaningful information about specific parameters and vehicle characteristics for immediate use by mission analysts.

The objectives of this study as specified in the contract are:

- To develop a technique in which stored data that is generated from a sophisticated launch vehicle trajectory optimization program may be speedily transformed into meaningful information about specific missions and vehicles for immediate use by the mission analysts and planners.
- To enable the analyst to economically and quickly generate new data (via interpolation and/or extrapolation within the data bank) on specified vehicles and missions.
- To provide an efficient method of cataloging and retaining previously compiled or newly generated performance information.
- To provide an efficient and concise retrieval system for trajectory/ performance data.

1.3 REPORT ORGANIZATION

In fulfilling specific objectives and expected results of the contract, most of the work tasks dealt with the detailed design and organization of the program logic based on known numerical techniques as opposed to basic research into devising new interpolation or extrapolation techniques. For this reason this final technical documentation of the work accomplished deals heavily with a precise description of the program design, how it works, and how it can be used. In Section II of the report a detailed discussion of the basic program capabilities is presented. In this discussion, the program objectives and the resulting designs of the basic routines are presented. In Section III, the numerical techniques used in the program are discussed. Only a brief discussion is given since lengthly documentations of the methods are available elsewhere. In Section IV, the results, conclusions, and recommendations derived from the study are presented. Section V is a bibliography of pertinent references. Appendix A contains an overall description of the computer program, while Appendix B gives a detailed description of the cataloging and storage methods. Appendix C presents a definition of required inputs and a description of how to set up the deck. Appendix D contains some sample computer runs and Appendix E is a program listing.

Section II

GENERAL DESCRIPTION OF PROGRAM CAPABILITIES

The Automatic Performance Trade-off (APT) computer program described in this report has the capability of quickly retrieving payload information for launch vehicles. This information is stored in large data banks in the program. The information is retrievable for various mission parameters (launch azimuth, orbital altitude, ascent profile) and vehicle parameters (thrust, ISP, and stage weight). The payloads are the optimum values for those particular sets of mission and vehicle parameters for which they are stored. These data are obtained from sophisticated launch vehicle optimization programs.

Information is also stored for determining a stage impact trace as a function of mission parameters. This information is also obtained from the launch vehicle optimization programs.

These data can quickly be retrieved from the data bank. By using interpolation and extrapolation techniques new information can be generated.

The program was designed so that the data banks could easily and quickly be modified. Appendix B describes the procedure for accomplishing this. Each vehicle, for which information is to be stored, has a separate data bank. This makes the program independent of the vehicle. Information can be stored for one, two or three stage vehicles. For information on the vehicles available as well as the specific data stored for each vehicle see Appendix C, Section 5.

The following sub-sections describe in detail the specific missions that the program can handle at present. A detailed discussion is given on the cataloging procedures used for storing data on the various missions.

The primary objective of the program is to determine performance effects on launch vehicles due to variations in mission parameters (e.g. orbital attitude, launch azimuth, and ascent profile) and in vehicle parameters (thrust, ISP, stage weight). The particular missions for which performance variations can be obtained are:

- Low Earth Orbits
  - Circular
  - Elliptical
- Polar Orbits
  - Circular
  - Elliptical
- High energy missions
- Synchronous orbits.

The polar orbits are not included in the low earth orbit category. This is because a yaw steering maneuver is used in achieving polar orbits. The cataloging techniques and the independent parameters for each mission are discussed in detail in the following subsections.

It will be noticed in the following discussion that the cut-off weight to a low earth orbit is the actual quantity stored. The program has the capability of giving the user the cut-off weight, gross payload or net payload.

2.1 PAYLOAD DETERMINATION FOR MISSION PARAMETER VARIATIONS

2.1.1 Low Earth Orbit

2.1.1.1 Circular Orbits. The low earth circular orbits are defined here to be those orbits whose altitudes range from 100 to 500 n mi. These orbits are achieved by direct injection with no yaw steering. For a given vehicle, the cut-off weight for circular orbits is a function of launch azimuth and orbital altitude.

Due to range safety restrictions, only the spectrum of launch azimuths of 45 to 145 degrees has been considered. It was found early in the study that for a fixed launch azimuth the variation in cut-off weight as a function of altitude can be represented with sufficient accuracy by only five data points. The data points which were selected are the cut-off weights for altitudes of 100, 200, 300, 400 and 500 n mi. Similarly, it was found that payload variation as a function of launch azimuth for a fixed altitude can also be represented by five data points. The data points selected are launch azimuths of 45, 65, 90, 120, and 145 degrees. Therefore, the basic technique used requires that 25 data points be stored to give adequate coverage for low earth circular orbit missions. Also stored are the five values of altitude and the five values of launch azimuth. Table 2-1 shows a typical example. Using the stored data, the cut-off weight for any desired launch azimuth and orbital altitude can be obtained by interpolation.

As a consequence of the interpolation method used, it is not necessary to have equal spacing between the values of the independent variables as can be seen in the selection of the launch azimuth points. Neither is it necessary that the data points be stored for the particular values of the independent variables cited above. It is important, however, that data points be stored for a launch azimuth of 90 degrees since this is approximately the azimuth at which the maximum cut-off weight is achieved.

At least five points in altitude and five points in azimuth should be used in setting up the basic cut-off weight matrix. It is necessary to reduce the error in interpolating within this matrix to a very low value. This is because this same matrix is used in determining cut-off weights for the elliptical orbits and high energy missions as will be discussed later.

The order of the interpolation used in a given direction is one less than the number of independent variable points for that direction. This n^{th} order interpolation is the largest possible order for n+1 points. For the example cited here, the order in each direction is four (4). By spacing the data as described, it was found that the error in interpolation was less than a tenth of one percent. Good results were also obtained for extrapolating short distances beyond the data bank.

2.1.1.2 Elliptical Orbits. The low earth elliptical orbits are constrained in altitude, launch azimuth, and ascent profile in the same manner as the circular orbits. To store data for this mission class naively using the same technique as was used for the circular orbits, would require the use of three independent variables. These independent variables are launch azimuth, perigee altitude,

Table 2-1. CUTOFF WEIGHT AS A FUNCTION OF LAUNCH AZIMUTH AND CIRCULAR ORBIT ALTITUDE

SATURN V TWO STAGE VEHICLE					
LAUNCH AZIMUTH (DEGREES)	CIRCULAR ORBIT ALTITUDE (n mi) 100	200	300	400	500
	CUTOFF WEIGHT (POUNDS)				
45	363886	323586	271788	216087	162728
65	372498	330749	277246	219924	165237
90	376720	334257	279906	221794	166447
120	371151	329633	276395	219332	164840
145	358873	319411	268593	213836	161240

and apogee altitude. However, after examining data for the cut-off weight as a function of these independent parameters a much simpler technique was devised.

For a fixed perigee altitude, the decrease in cut-off weight is a linear function of apogee altitude for the altitude range previously specified. Figure 2-1 shows the lines for several different values of perigee altitude. It was found that the slopes of these lines did not change as a function of launch azimuth. Therefore, the only data points stored for this problem are the slopes of these lines.

The slope values are stored for the altitude values used in the circular orbit case. A zero is used for the slope value corresponding to the highest altitude used. The order of interpolation is set to one less than the number of altitude points stored.

The following procedure is used to determine the cut-off weight to any low elliptical earth orbit of a given perigee and apogee altitude. The first step is to determine the cut-off weight (CSC) to a circular orbit or perigee altitude. This is done using the cut-off matrix described previously. The second step is to determine the appropriate slope (SLOPE) value for the given perigee altitude. This is done by interpolating the slope and perigee altitude arrays. The desired cut-off weight (CWE) is then obtained by:

$$CWE = CWC + SLOPE\ (ALTA-ALTP)$$

where ALTA = apogee altitude

ALTP = perigee altitude

2.1.1.3 Perigee Injection into an Elliptical Orbit and Circularizing at Apogee. The program will make an estimate of the cut-off weight for perigee injection into a low elliptical earth orbit and then circularizing at apogee. This estimate is based on the mass change calculated by the characteristic velocity equation. To do this the program first determines the cut-off weight CWE for the specified elliptical orbit using the procedure described in subsection 2.1.1.2. The payload for circularizing at apogee CWEC is then obtained by:

$$CWEC = CWE\ EXP\ [-\Delta V/ISP*G]$$

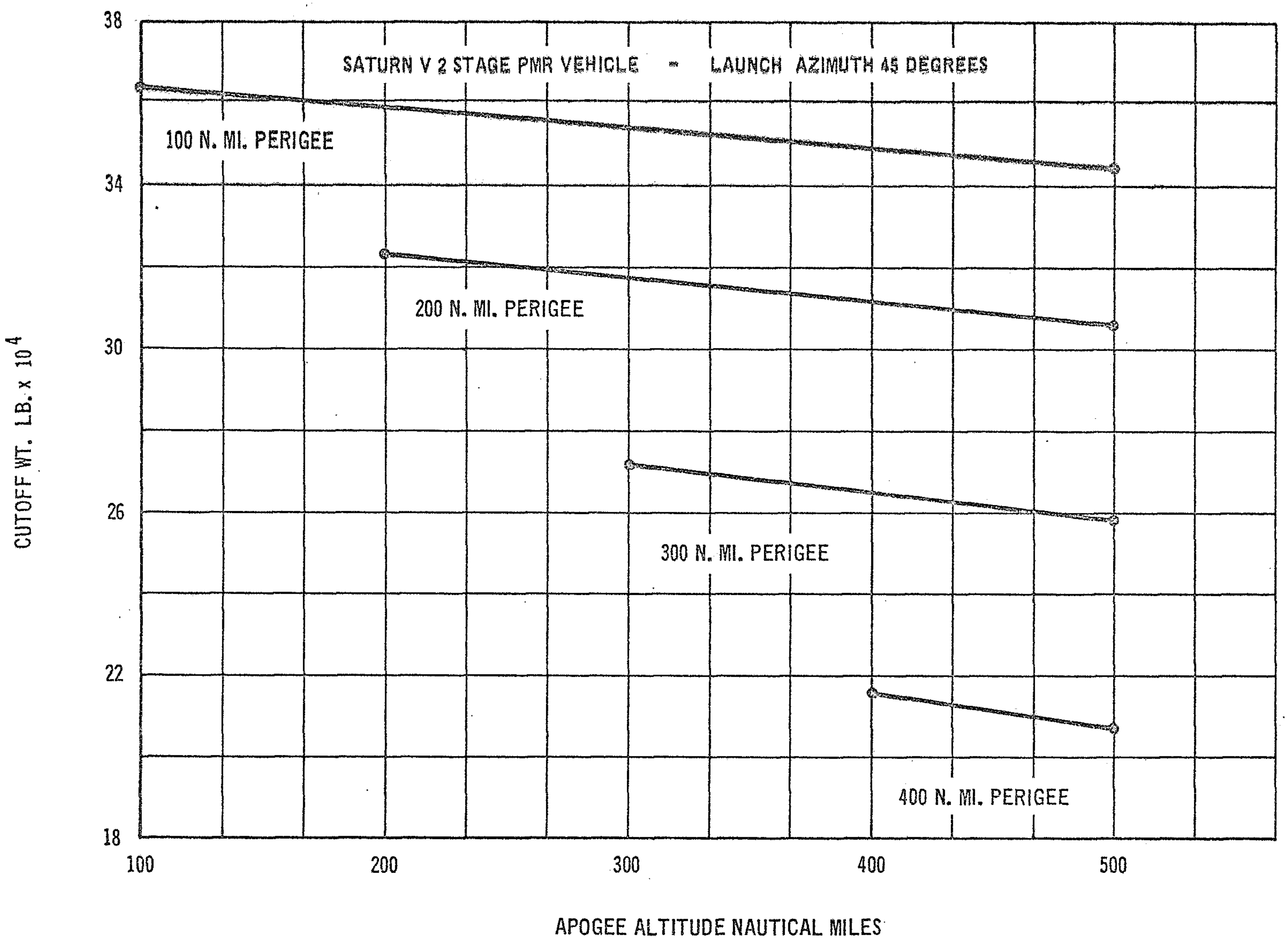

Figure 2-1. VEHICLE PERFORMANCE TO PERIGEE INJECTION VERSUS APOGEE ALTITUDE

where ΔV = Required circularizing velocity change

ISP = Specific impulse

G = Gravity constant.

2.1.2 Polar Orbits

Polar orbits are achieved by executing a yaw steering maneuver at some time, t, after lift-off. This yaw maneuver continues at a fixed rate until polar orbit is achieved. Only launch azimuths of 45 and 145 degrees were considered for this problem. Two separate data banks are used; one corresponding to the launch azimuth of 45 degrees and the other for the 145 degree launch azimuth. The program will not interpolate between azimuths for this problem.

2.1.2.1 Circular Polar Orbits. The range of altitudes considered for circular polar orbits was from 100 to 500 n mi. The program is equipped to handle three different yaw steering times. The rate of yaw steerings is that rate which will produce the maximum payload. The cut-off weight matrix is then generated as a function of altitude and time of yaw steering initiation. This cut-off weight matrix is constructed like the one shown in Figure 2-1 for circular orbits with no yaw steering. This matrix is stored along with the particular values of altitude and yaw steering time. Then the cut-off weight for any altitude and yaw time can be obtained by interpolation.

2.1.2.2 Elliptical Polar Orbits. The range of perigee and apogees for elliptical orbits has the same range as the circular orbits. For fixed yaw time and perigee altitude the cut-off weight was found to vary linearly with apogee altitude. Figure 2-1 shows a typical example for the low elliptical orbits with no yaw steering. The same characteristics of those curves apply here. It was found that the slopes of these lines were independent of the time that yaw steering began. Therefore, only five points have to be stored. These slope points correspond to the altitudes used for circular orbit. A zero is input for the slope value at the highest altitude. The order of interpolation used is one less than the number of altitude points stored.

The following procedure is used to determine the cut-off weight to an elliptical polar orbit with a given perigee and apogee. This is for a fixed launch azimuth and yaw steering time. The first step is to determine the cut-off weight (CWPC) to a circular orbit with perigee altitude as previously discussed. Then the slope (SLOPEP) corresponding to the given perigee altitude is determined. The desired cut-off weight (CWPE) is then obtained by:

$$CWPE = CWPC + SLOPEP\ (ALTA - ALTP)$$

where ALTA = apogee altitude
ALTP = perigee altitude.

2.1.3 High Energy Missions

High energy missions are those where the required Vis Viva energy constant, C_3, (twice the total geocentric energy per unit mass) has been specified along with a right ascension and declination of the outgoing asymptote. Interplanetary missions are typically specified in this manner. The C_3 missions considered by the program first proceed to a low circular earth orbit. It is assumed that the absolute value of the declination of the outgoing asymptote $|\delta|$ is sufficiently small to allow a planer C_3 insertion burn. Under this assumption and with proper timing, the engines are reignited and continue to burn until the vehicle has achieved the designated C_3 value. Thus, as far as the APT program is concerned, a C_3 value completely specifies the out of orbit phase of a high energy mission.

Therefore, for a fixed launch azimuth and parking orbit altitude, the payload was investigated as a function of C_3. It was found that the fraction of the parking orbit cut-off weight remaining when the desired C_3 was achieved is a smooth function of C_3 (Figure 2-2). This curve was found to be independent of launch azimuth. The program stores this curve for a parking orbit altitude of 100 n mi.

The program uses the following procedure to determine the cut-off weight to any C_3 for a given launch azimuth. First, the data stored for low earth circular orbits are used to determine the cut-off weight (CWC) to

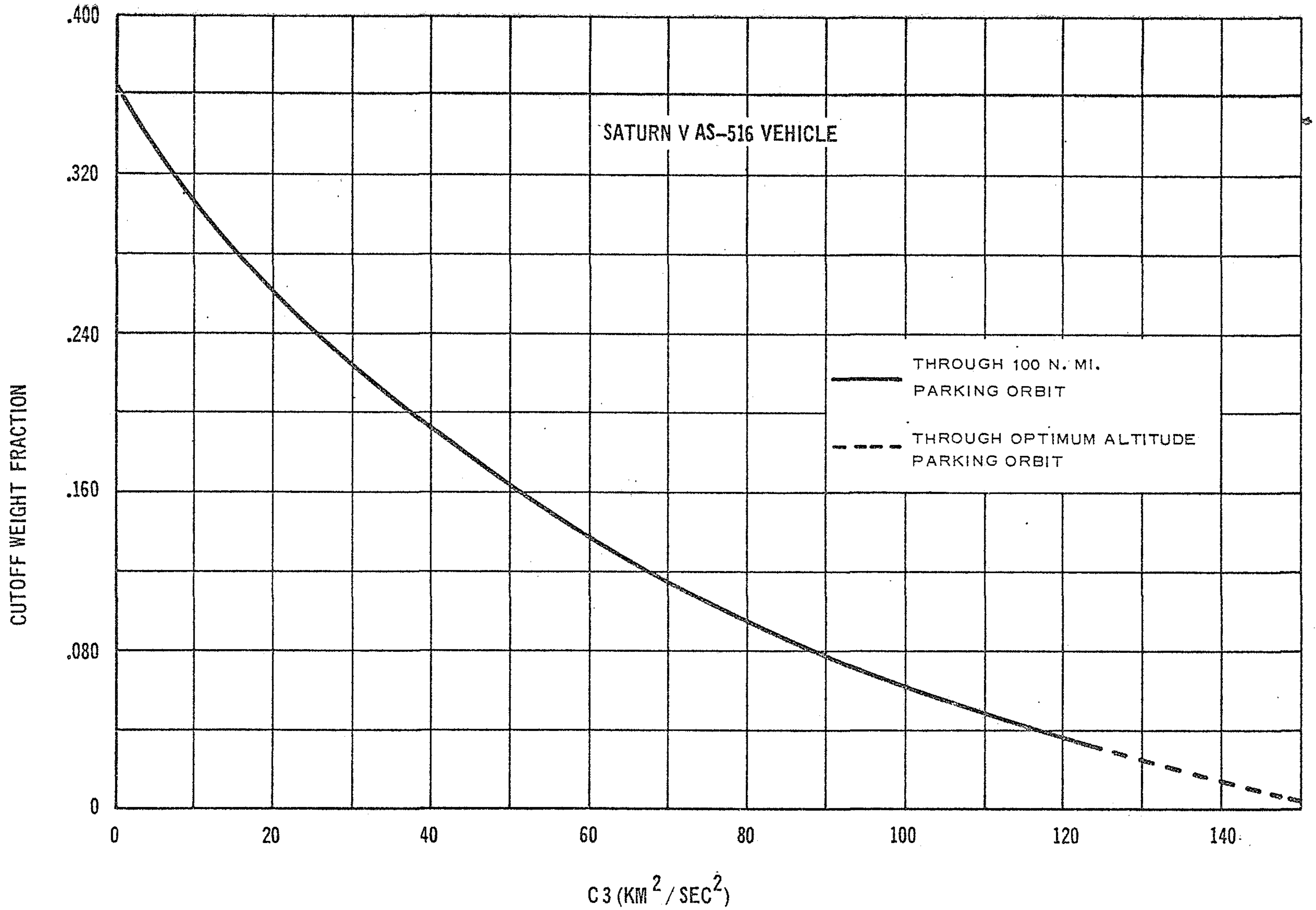

Figure 2-2. FRACTION OF PARKING ORBIT CUT-OFF WEIGHT REMAINING AFTER BURNING TO A GIVEN C3

parking orbit. Then the weight fraction $G(C_3)$ is obtained from the above mentioned curve for the designated C_3. The desired cut-off weight at C_3 insertion (CWC_3) is then

$$CWC_3 = CWC\ G(C_3).$$

On some of the larger vehicles considered it was found that for energy levels greater than a certain C_3 value the altitude of the parking orbit changed. This change in altitude is necessary to increase the final payload. Stored in the program are two separate weight fraction curves. One curve corresponds to C_3 values before the critical C_3 value and the other curve corresponds to C_3 values equal to and greater than the critical value. Data are also stored so that the optimum parking orbit altitude can be calculated for C_3 greater than the critical value.

2.1.4 Synchronous Orbit

The program is capable of handling payload information for both the earth equatorial synchronous orbit and inclined orbits with synchronous orbit altitudes. All of these orbits proceed first by direct injection with no yaw steering to a 100 n mi circular parking orbit. Besides calculating a payload to the final orbit, the program can perform an additional service for this mission. The program will calculate launch azimuth opportunities so that a specified longitude (hover point) will be obtained for the final orbit. The number of coasts and the time in orbit is calculated for the respective launch azimuths.

The cut-off weight to a 100 n mi circular parking orbit is stored as a function of launch azimuth. The spectrum of launch azimuths range from 45 to 145 degrees. The cut-off weight to parking orbit for any launch azimuth is then obtained by interpolation. The cut-off weight (CWSO) in the final orbit is obtained from the cut-off weight (CWPO) to parking orbit by:

$$CWSO = CWPO\ EXP[-\Delta V/ISP\ G]$$

where ΔV = required velocity change

ISP = specific impulse

G = gravity constant.

The appropriate velocity changes are obtained assuming a Hohmann transfer between orbits.

For the earth equatorial synchronous orbit, a 2.5 degree plane change was performed leaving parking orbit. The remaining plane change is removed at apogee. This maneuver results in obtaining the maximum payload in the final orbit. The required ΔV is given by:

$$\Delta V = \sqrt{\left(\frac{K}{R1}\right)^2 - \frac{\mu}{R1} - \frac{2K}{R1}\sqrt{\frac{\mu}{R1}}\cos(2.5°)} + \sqrt{\left(\frac{K}{R2}\right)^2 - \frac{\mu}{R2} - \frac{2K}{R2}\sqrt{\frac{\mu}{R2}}\cos(i-2.5)}$$

where R1 = radius to 100 n mi parking orbit (KM)

R2 = radius to synchronous orbit altitude (KM)

μ = 39860.2 KM^3/SEC^2

i = inclination of parking orbit (degrees)

$K = \sqrt{\frac{2\mu\ R1R2}{R1 + R2}}$

For the inclined orbit with synchronous orbit altitude no plane change is required. Therefore, both the angles in the above ΔV equation are zero. So,

$$\Delta V = K\left[\frac{1}{R1} - \frac{1}{R2}\right] + \sqrt{\frac{\mu}{R1}} - \sqrt{\frac{\mu}{R2}}$$

where the variables are defined as before.

From actual runs the exponential term in the equation is determined. This number is then used as a correction factor on the theoretical value. A separate number is obtained for both types of orbits. These numbers are stored along with the cut-off weight for the particular vehicle. These numbers are computed for a launch azimuth of 90 degrees and assumed to hold over the spectrum of 45 to 145 degrees.

When a longitude is specified for the final orbit, the program will calculate the launch azimuth opportunities and determine the number of coasts required in their corresponding 100 n mi parking orbit. Departure from the parking orbit occurs at either the ascending or descending nodes. Also determined is the time from injection into the parking orbit to the first descending node and the time from this node to departure. The data used for this portion of the program were taken from reference 2.

2.2 PAYLOAD DETERMINATION FOR VARIATIONS IN VEHICLE PARAMETERS

Each vehicle is described in its respective data block by three parameters for each stage. These three parameters are initial thrust level, specific impulse, and dry stage weight. The effects of performance variations due to changes in any of these parameters are predicted by psuedo-partial derivatives of payload with respect to each parameter. These psuedo-partial derivatives are called exchange ratios.

The exchange ratio for a particular vehicle parameter is obtained by varying this vehicle parameter and holding all other mission and vehicle parameters constant. The value of the exchange ratio is then the variation in payload divided by the change in vehicle parameter (Figure 2-3). The variations in the vehicle parameters are kept small so that the resulting variation is payload in linear.

Exchange ratios are stored as a function of altitude for low circular earth orbits and a function of C_3 for high energy missions.

The exchange ratios also have to be determined as a function of vehicle stage. These exchange curves are found to be independent of launch azimuth.

These exchange ratios are then stored as a function of stage and mission parameter. The corresponding mission parameter values are also stored.

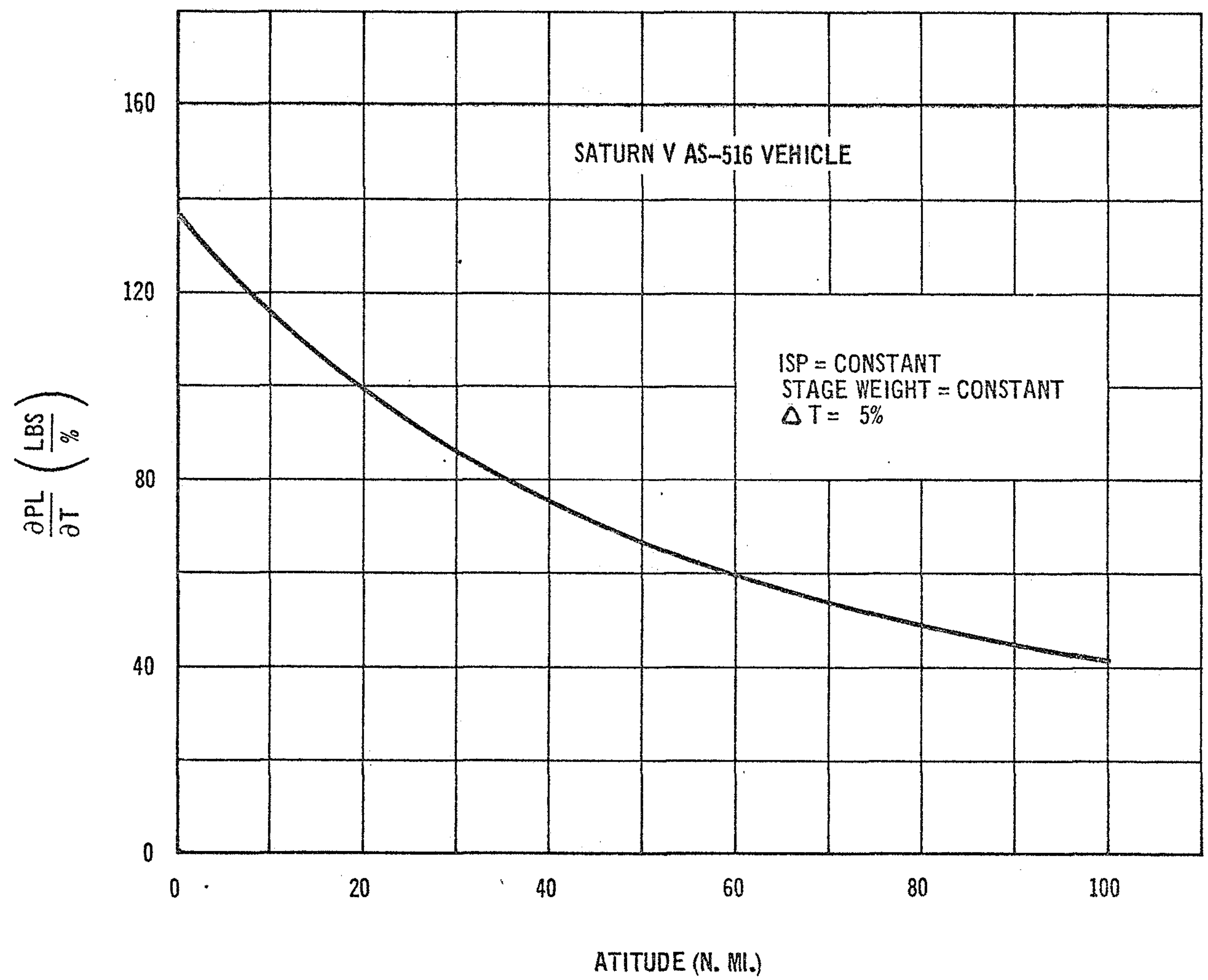

Figure 2-3. EFFECT OF TOTAL FIRST STAGE THRUST ON PAYLOAD VERSUS C_3

Interpolation or extrapolation can then be used to determine the exchange ratio for any specified stage at any altitude or energy. The payload change for a given amount of change in the vehicle parameter equals the product of the exchange ratio times the amount of change.

Care should be taken in using these exchange ratios. The limits indicated for each variable should not be exceeded. The variations in payload can only be assumed linear between the given limits.

2.2.1 Thrust

The variations in thrust were ± 5 percent of the nominal value on each stage. For those stages with programmed thrust level changes as a function of time, the entire thrust curve was increased by 5 percent.

2.2.2 Specific Impulse

The variations in ISP were ± 5 seconds in the first stage and ± 10 seconds in the upper stages.

2.2.3 Stage Weight

The variations in stage weight were ± 5 percent for the lower stages. For the last stage this exchange ratio is -1. Thus, for each pound of structure weight there is a loss of one pound of payload.

2.3 STAGE IMPACT TRACE

It is necessary to know the point that the vehicle would impact the earth if the engines failed before orbital or escape velocity is obtained. The program can store sufficient data to determine an impact point trace as a function of engine failure time for the various missions. Two separate methods are used for handling: (1) the cases with no yaw steering and (2) those with yaw steering (Polar Orbits). The program presently will handle only impact traces for variations in mission parameters using nominal values of the vehicle parameters.

For a given set of mission parameters, the latitude and longitude of the stage impact point as a function of time was obtained from the ROBOT deck (ref. 10). These latitude and longitude points as a function of time after lift-off were then curve fitted separately using the method of least squares. The resulting coefficients were then stored in the program. It was found that a sixth-order polynomial is a good fit for both the latitude and longitude points. By using this method, the number of points that had to be stored was reduced considerably.

The ROBOT deck has the capability of calculating the impact point of a vehicle on a spherical earth. This calculation is made at each time step in the integration of the ascent trajectory. The program is also capable of integrating the booster's trajectory back to the ground. Integrating the trajectories back would require an unacceptable number of runs to obtain sufficent information for all missions. The results from the two method were sufficiently close so that the predictions by the former method were used.

Since a spherical earth is used in calculating the impact points, a single method was devised to handle the traces as a function of launch azimuth. This method is good for trajectories where no yaw steering is used. For a given set of mission parameters the coefficients of the latitude and longitude curve fits are stored. These points are taken for a launch azimuth of 90 degrees. The impact trace corresponding to any other launch azimuth can be obtained by rotating the 90 degree trace about an axis through the launch site to the center of the earth. The angle through which the trace is rotated is equal to the difference in the two launch azimuths.

The following rotations are required to obtain the impact trace for any launch azimuth from the 90 degree trace, see Figure 2-4.

$$\underline{X}^{I} = \underline{\underline{T}}_{Z}(\phi(1))\underline{X}$$

here

$\underline{X}$ = Column matrix for the components of the impact point for the 90 degree trace at any time t.

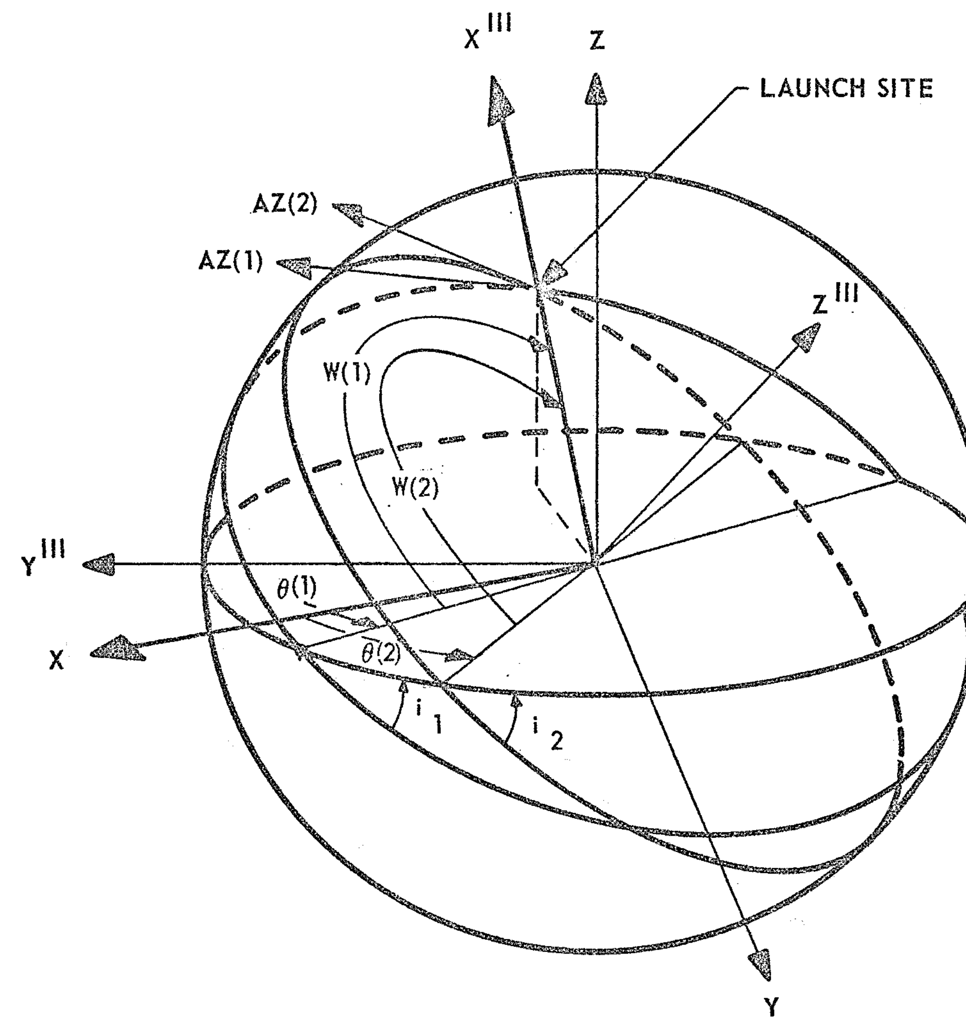

Figure 2-4. ANGLES USED IN DETERMINING TRANSFORMATION MATRIX FOR CALCULATING STAGE IMPACT TRACES

$\underline{\underline{T}}_Z(\theta(1))$ = Rotational matrix about the Z-axis through an angle $\theta(1)$.

(1) = refers to angles associated with the 90 degree trace.

and

$$\underline{X}^{II} = \underline{\underline{T}}_X(-i(1))\underline{X}^{I}$$

$$\underline{X}^{III} = \underline{\underline{T}}_Z(-w(1))\underline{X}^{II}$$

$$\underline{X}^{IV} = \underline{\underline{T}}_Z(w(2))\underline{X}^{III}$$

$$\underline{X}^{V} = \underline{\underline{T}}_X(i(2))\underline{X}^{IV}$$

$$\underline{X}^{VI} = \underline{\underline{T}}_Z(-\theta(2))\underline{X}^{V}$$

where (2) = refers to angles associated with the desired launch azimuth.
Therefore the desired impact point is found by:

$$\underline{X}^{VI} = \underline{\underline{T}}_Z(-\theta(2))\ \underline{\underline{T}}_X(i(2))\ \underline{\underline{T}}_Z(w(2))\ \underline{\underline{T}}_Z(-w(1))\ \underline{\underline{T}}_X(-i(1))\ \underline{\underline{T}}_Z(\theta(1)\underline{X}$$

The values of i(1) and $\theta(1)$ are 28.37 and 9.43 degrees respectively. The value for w(1) was found by rotating the launch sight points on to the X^{III} coordinate system and taking the angle between the resulting X and Y components.

Referring to Figure 2-4, it can be seen that:

$$\underline{\underline{T}}_Z(-\theta(2))\ \underline{\underline{T}}_X(i(2))\ \underline{\underline{T}}_Z(w(2)) = \underline{\underline{T}}_Z(-\theta(1)\ \underline{\underline{T}}_X(i(1))\ \underline{\underline{T}}_Z(w(1))\ \underline{\underline{T}}_Z(AZ-90)$$

where AZ = desired launch azimuth.

Therefore the desired impact point can be found by:

$$\underline{X}^{VI} = \tilde{\underline{\underline{A}}}\ \underline{\underline{T}}_X(AZ-90)\ \underline{\underline{A}}\ \underline{X}$$

where

$\underline{\underline{A}} = \underline{\underline{T}}_Z(-w(1))\ \underline{\underline{T}}_X(-i(1))\ \underline{\underline{T}}_Z(\theta(1))$

$\tilde{\underline{\underline{A}}}$ = transpose of $\underline{\underline{A}}$

The matrices $\underline{\underline{A}}$ and $\underline{\underline{\tilde{A}}}$ are stored in the program and only the matrix $\underline{\underline{T}}_X$ (A_Z-90) has to be computed each time. The results obtained using this method are in excellent agreement with data taken from actual ROBOT runs.

2.3.1 Low Earth Orbits

2.3.1.1 Circular Orbits. The low earth circular orbits are achieved by direct injection with no yaw steering. The coefficients of the latitude and longitude curves are stored as a function of orbital altitude for a launch azimuth of 90 degrees. The altitudes usually correspond to those used in the payload determination. The trace for any launch azimuth can be obtained using the previously described method. Interpolation can be used to get the trace for any altitude.

2.3.1.2 Elliptical Orbits. These orbits are also obtained by direct injection with no yaw steering. It was found that the impact traces for an elliptical orbit for a fixed perigee varied only slightly from the one for a circular orbit with perigee altitude. Therefore the elliptical impact traces are handled by using the data for the circular orbits. The program interpolates on altitude to obtain the correct perigee altitude and rotates the trace to the designated launch azimuth.

2.3.2 Polar Orbits

2.3.2.1 Circular Orbits. A different technique from the one described previously is used to determine the impact traces for polar orbit. Since a yaw steering maneuver is used during the ascent an additional variable must be considered. The coefficient for the latitude and longitude curve fits are stored as a function of circular orbital altitude and the time yaw steering began. The impact point for any time is then obtained by interpolation.

2.3.2.2 Elliptical Orbits. The variation between the impact traces to an elliptical polar orbit and to a circular polar orbit of perigee altitude was very slight. Therefore the traces to elliptical polar orbits are calculated using the data for circular polar orbits. The altitude is interpolated to correspond to the designated perigee value.

2.3.3 High Energy Missions

All the high energy missions considered proceed first through a circular parking orbit. The program will presently handle an impact trace to a 100 n mi parking orbit. The coefficients of the latitude and longitude curve fits are stored as a function of the energy. These traces are obtained for a launch azimuth of 90 degrees. The impact trace for any launch azimuth is obtained by rotating the stored trace as described previously. The change in the trace for a given energy is obtained by interpolation.

2.3.4 Synchronous Orbits

All synchronous orbits considered by the program proceed through a 100 n mi circular parking orbit. The coefficients of the latitude and longitude curves are stored for a launch azimuth of 90 degrees. The variation in the impact trace due to a change in launch azimuth is handled by rotating this trace as described previously.

Section III

METHOD OF INTERPOLATING AND EXTRAPOLATING

3.1 BASIC METHOD

The Lagrange method of interpolating was selected for the program. One of the major reasons for selecting this method was that it does not require equal spacing between consecutive values of the independent variable. This feature also allows the Lagrange method to be readily used for inverse interpolation. The method was easily programmed and extended to handle several independent variables. The value of the dependent variable obtained is independent of which independent variable was interpolated for first.

This method fits an n^{th} degree polynomial through the n+1 points formed by the dependent and independent variables. The resulting polynomial approximates the function through the data points and can be used for interpolation and extrapolation. A derivation of this formula can be found in most numerical analysis books, see references 3, 4, or 6.

3.2 MULTI-DIMENSIONAL INTERPOLATION

A multi-dimensional interpolation subroutine was developed which can handle up to four independent variables. The routine is supplied with independent and dependent variable arrays, the number of points in each array, the order of interpolation desired, and the values of the independent variable for which a result is to be obtained. The largest order of interpolation allowable for n points is n-1. The program is designed to use Lagrange's method to interpolate or extrapolate for a particular value of the dependent variable. There is also a feature such that the coefficients for some polynomial can be supplied to this routine through the dependent variables array. The desired dependent variable is then obtained by raising the given value of the independent variable to the appropriate powers and multiplying them by the corresponding coefficients.

The routine will handle a combination of the above methods. In doing this the coefficients of the curve fit must be the first independent variable supplied. See Appendixes B and C for description of the data storage system and input description, respectively. This was developed to handle stage impact traces. The impact points are curve fitted as a function of engine failure time and the coefficients are stored as a function of mission parameters such as altitude or energy.

3.3 FIGURE OF MERIT

As previously discussed, payload to any altitude or azimuth is computed by interpolating and extrapolating in a stored array of data. Empirical results have shown that the Lagrange interpolation formula can determine payloads within the array that are accurate to 0.1 percent or less. This error is small enough to disregard. This accuracy is obtained by using the largest order of interpolation possible.

Empirical results have also shown that the Lagrange Formula when used to extrapolate outside the array produces errors as great as six percent. This error occurred at one spacing outside the data bank in the altitude direction. The spacing in altitude is 100 n mi. Payloads computed by extrapolation are only acceptable when the error limits can be predicted to a tolerance of ± 2 percent. The standard Lagrange remainder term cannot be computed because the higher derivative of payload with respect to azimuth or altitude is not known. Several techniques were derived to predict this error and establish a figure of merit. The most favorable of these schemes is described below.

A grid of payloads as a function of launch azimuth and circular orbital altitude was generated using a two-stage Saturn V vehicle. The altitude ranged from 100 to 500 n mi with an average delta of 100 n mi. Launch azimuth ranged from 45 to 145 degrees with an average delta of 20 degrees. The mean error encountered in extrapolating to an altitude of 600 n mi for all azimuths was found to be 5.5 percent. Similarly, the extrapolation error for azimuths of 0 to 180 degrees was found to be 1 percent.

The figure of merit is computed assuming that the error increases linearly in each of the two independent variables. The empirical data mentioned above were used to determine the slopes for the error curves. The total error is thus the sum of the two errors. This value is presented as the figure of merit and is printed only when the predicted value exceeds 2 percent.

Section IV

RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

4.1 RESULTS AND CONCLUSIONS

The principal result of the study was the design, construction, and debugging of the Automatic Performance Prediction (APT) computer program. The successful development of the program satisfies the study objectives given in subsection 1.2 in every respect.

From the study, it has been shown that excellent results can be obtained from a computer program of this nature. The program performs well in determining both payload information and stage impact traces. The cataloging techniques developed for the various missions will be extremely useful in generating data for a new vehicle.

The study showed that data from optimization programs could be stored and numerical methods used to interpolate or extrapolate for new data. The error due to interpolation for payload as a function of launch azimuth and circular orbital altitude was consistently less than a tenth of a percent. The speed with which the program performs makes it a very useful tool for preliminary mission studies. The program has the capabilities of being expanded in the future to handle problems in different areas. By systematically studying the problems in related areas, it appears that similar techniques can be used for storing and retrieving useful information.

4.2 RECOMMENDATIONS

The following recommendations are made for increasing the usefulness of the program to the mission planner. These recommendations, if implemented, will give the planner the capability of faster response to problems from conception to the final reports.

- Develop into the program the capability to call sophisticated launch vehicle optimization programs. Thus, when searching for information which lies at a large distance outside the data bank, the optimization programs would be called. Runs could then be made using the optimization decks that would increase the data bank to the appropriate size. The appropriate size would be where the error encountered by interpolation and extrapolation would be acceptable. This information generated would then be stored for future use.

Another use for this feature would be with new vehicles. Here the appropriate vehicle data could be supplied along with the types of missions to be considered. A data spectrum would then be generated for a given mission in accordance with the cataloging techniques previously defined. These data could then be stored for future use. This option would reduce human error to a minimum. The gain in efficiency would greatly increase the mission planner's response capability for preliminary studies.

- It is recommended that a conversational mode be added to the program. This feature would allow the user to communicate with the program and would allow the program to be used by more people by eliminating the time required to study program documentation.

  With a conversational mode the program would proceed in the following manner. Initially, the program would display the program's major options in sequence to the user. The user could select the desired options by input through a typewriter. After selecting the various options, the program would then display the corresponding information stored for the various vehicles. The user would then select the appropriate vehicles. The program, stepwise, would show the user how to input values for the mission and vehicle parameter cases for which data are desired. The program would then determine the desired numbers by interpolating and extrapolating. The program would then display output options which are available. This process could be repeated until the user has solved his particular problem. This type of mode could also be connected to the storage banks. Then the user could be given detailed instructions in how to store his data for future use.

- At present, the program only outputs using a standard printer. The incorporation of the use of other types of output devices would enhance the usefulness of the program. The addition of plotting routines could be used to show trends in vehicle performance. They could also be used to show impact trace on a graph of the earth. The use of an oscilloscope output would be helpful in connection with the conversational mode. The output mode would also allow the user to see his results before having them printed.

Section V

REFERENCES AND BIBLIOGRAPHY

1. Brooks, R.M., "Earth Synchronous Orbit Study," FR-66-309, Huntsville, Alabama, Raytheon Company, October 1966.

2. Ellison, B., "Preliminary Mission Analysis Aid for the Zero Inclination Earth Synchronous Orbit," R-AERO-IN-5-66, Huntsville, Alabama, George C. Marshall Space Flight Center, May 1966.

3. Kunz, K.S., Numerical Analysis, New York, McGraw-Hill Book Company, Inc., 1957.

4. Nielsen, K.J., Methods in Numerical Analysis, New York, Macmillian Company, 1964.

5. Pennington, R.H., Introductory Computer Methods and Numerical Analysis, New York, Macmillian Company, 1965.

6. Scarborough, J.B., Numerical Mathematical Analysis, Baltimore, The John Hopkins Press, 1962.

7. Toelle, R.G., "A Performance Study for the Application of the Saturn V to High Energy Earth Escape Mission," X-53639, Huntsville, Alabama, George C. Marshall Space Flight Center, July 1967.

8. White, J.F., Flight Performance Handbook for Powered Flight Operations, New York, John Wiley and Sons, Inc., 1963.

9. IBM 7090/7094 Programming Systems FORTRAN IV Language, New York, IBM Systems Reference Library, 1964.

10. Gottlieb, R.G., "ROBOT - Apollo and AAP Preliminary Mission Profile Optimization Program," Huntsville, Alabama, Applied Analysis, Inc., December 1967.

Appendix A

COMPUTER PROGRAM DESCRIPTION

A.1 GENERAL

The program was written using FORTRAN IV language. It requires a machine of at least 32,000 words of core and the system must have an overlay feature. The program presently accepts card input.

A.2 MACRO FLOW CHART

Figure A-1 shows the various options available in the program. In each block is the subroutine name which corresponds to the various options.

A.3 DATA BANK

Through the subroutine BANK, the appropriate data bank is loaded into core. Figure A-2 is a diagram depicting how the data banks are linked to the program.

The data stored by the program is in BLOCK DATA subroutines. Each BLOCK DATA routine contains all the information on a particular vehicle. Since only one set of vehicle data is needed in core at any one time, the BLOCK DATA routines are overlayed.

Only executable subroutines or functions may be referenced in an overlay structure. Therefore, the following technique was used to assure loading of the desired block data routine. Behind each block data routine is placed a reference subroutine which causes no program action but does define an overlay origin point. The main program executes a call to the appropriate vehicle reference subroutine. This call causes the information in the block data routine to be loaded with the overlay segment defined by the reference subroutine.

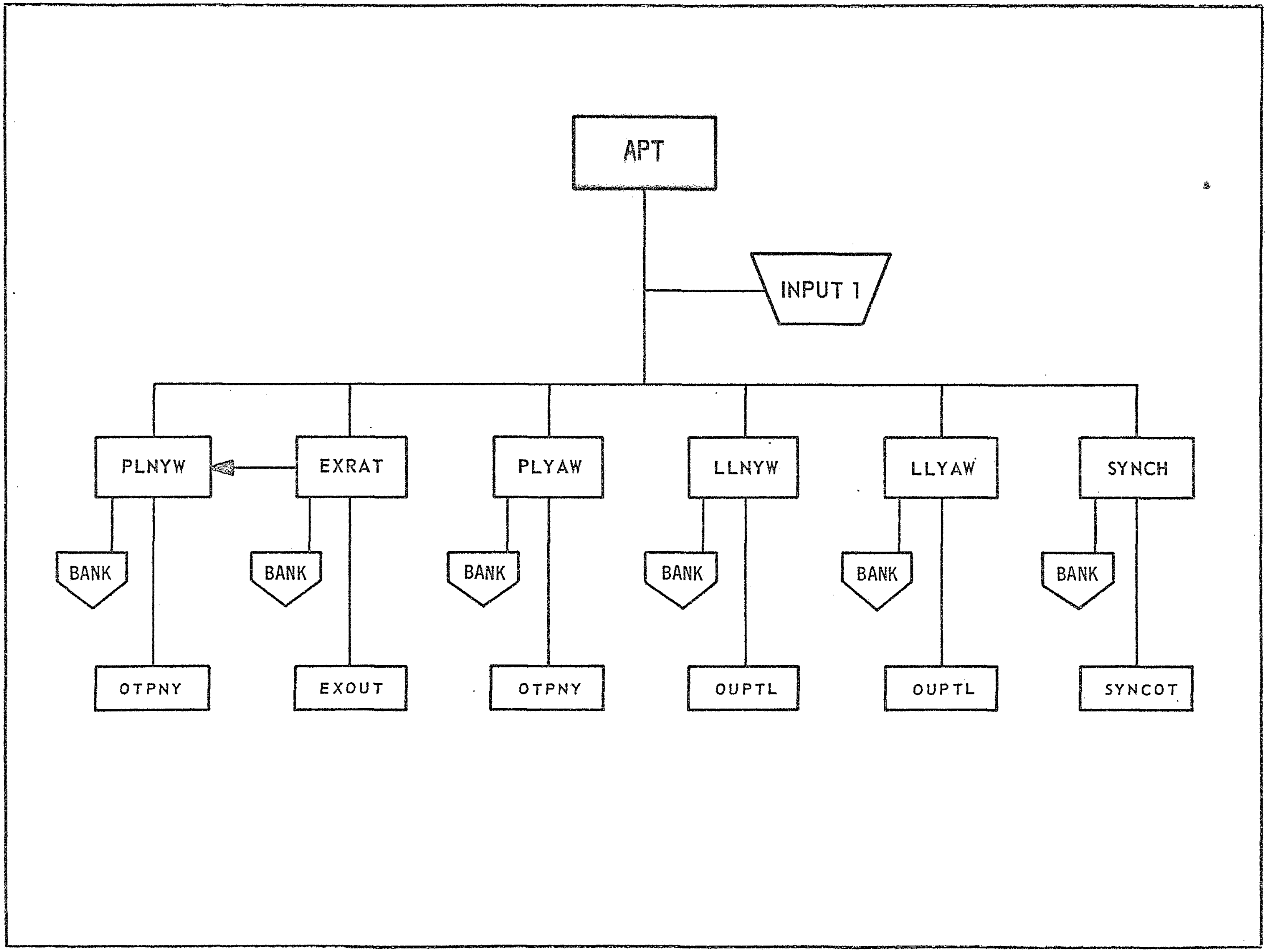

Figure A-1. MACRO FLOW CHART OF APT

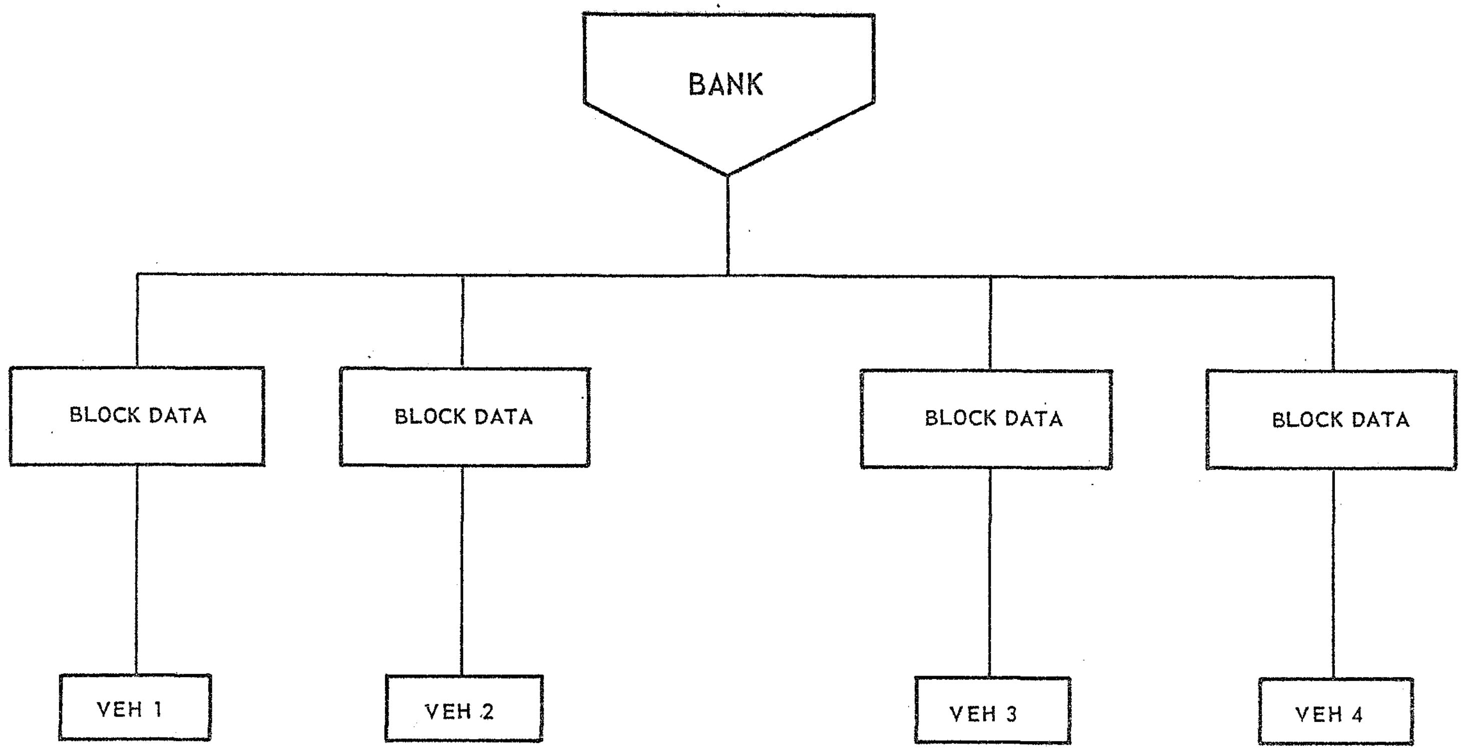

Figure A-2. MACRO FLOW CHART OF DATA RETRIEVAL

Appendix B

DATA STORAGE METHOD DESCRIPTION

B.1 GENERAL

The cataloging techniques described previously determine what information will be stored for a given mission. The information making up a data bank is sorted in BLOCK DATA subroutines. By storing the data in this manner it was not necessary to develop programs to write and read data from a tape. With this method it is very easy to make corrections and additions to the data banks.

A BLOCK DATA routine contains all the information to be stored for one particular vehicle.

B.2 INSTRUCTIONS

The following dictionary gives a brief description of each FORTRAN variable under which data are stored in the data banks. Beside each FORTRAN variable are the actual mission parameters upon which it is functionally dependent. Also listed is the maximum dimension allowed for the variable. All the FORTRAN variables are singly subscripted, for easier program handling of arrays. The variable type is also indicated. An F designates floating point, an I designates integer, and an A designates alphameric.

Consider the data for the dependent variable to be represented by a large matrix, A. Let the independent variable arrays be X1, X2, and X3. Here X1, for example, could represent altitudes of 100 to 500 n mi. Then the data for A(X1, X2, X3) is stored for the first independent variable (here X1) varying the fastest and the last independent variable (here X3) varying the slowest.

To illustrate how data are input, consider the first set of variables described in the dictionary which follows. These variables are NX, NORX, X, and F1. Assume that there had been 6 launch azimuths and 3 altitudes. The X array would then consist of 6 values of launch azimuth followed by three values of altitude. The variables are input in increasing order. The value of NX(1) and NX(2) would be 6 and 3, respectively. The product of NX(1) with NX(2) must be less than or equal to 25. The product of the number of points of each of the independent variables must be less than or equal to the dimension for the dependent variable array. The order of interpolation could be 5th and 1st, respectively. Thus set NORX(1) = 5 and NORX(2) = 1. The values for the dependent variable are input in accordance with the above statement. Here azimuth and altitude correspond to the first and second subscripts respectively. For our examples the first six values of F1 would be the cut-off weights of the first altitude values for the six values of launch azimuth. The next six values would correspond to the second altitude and all launch azimuths. This process is continued until the values for all altitudes are input. For this example there would be 18 values of F1 corresponding to the six azimuth and three altitude values.

To store data for the stage impact traces the following rules must be followed. The first independent variable is the time of engine failure. The coefficients of the impact points latitude and longitude curve fits are input for increasing powers of time. For example let the latitude = $a_o + a_1 t + a_2 t^2 + a_3 t^3 \ldots a_6 t^6$. Then the coefficients are input in the following order a_o, a_1, a_2 .... a_6.

The input has been divided into several sections indicated by horizontal bars. Above each section is a brief description of the mission for which the data is being stored. All information must be input for the program to operate properly.

If the maximum order of interpolation is desired for a particular parameter, set the appropriate FORTRAN variable for order equal to one less than the number of points for that parameter.

B.3 DATA STORAGE INPUT DICTIONARY

Variable Name	Function	Maximum Dimension	Description
			Payload Determination for Low Circular Earth Orbits
NX		2	(1) Number of launch azimuths used for payload determination . (I) (2) Number of circular orbit altitudes.
NORX		2	(1) Order of interpolation in the azimuth direction. (I) (2) Order of interpolation in the altitude direction.
X	(Azimuth, Altitude)	(10)	Independent variable array for payload determination for mission parameters. The first NX(1) values correspond to azimuth and the next NX(2) values correspond to altitude. (F)
F1	(Azimuth, Altitude)	(25)	Dependent array of cut-off weights to circular orbits. (F)
			Payload Determination for Low Elliptical Earth Orbits
NXE			Number of points in the SLOPE array. (I)
NORE			Order of interpolation for SLOPE. (I)
SLOPE	(Perigee Altitude)	(5)	Slopes of the payload versus apogee altitude lines as a function of perigee altitude. Perigee altitude should correspond to altitudes used in X array. (F)
			Stage Impact TRACE for Low Circular Earth Orbits*
NXLLP		(3)	(1) Number of coefficients stored for the stage impact point versus engine failure time. Set equal to 7. (I) (2) Number of altitudes.
NORLLP		(3)	(1) Set equal to 7. (I) (2) Order of interpolation for altitude direction.
XLLP	(Engine cut off time altitude)	(16)	Input 7 zeroes followed by NXLLP(2) values of the altitudes. (F)
F2	(Engine cutoff time, altitude)	(35)	Coefficients of the impact point latitude versus engine failure time curve fit. (F)
F3	(Engine cut-off time, altitude)	(35)	Coefficients of the impact point longitude versus engine failure time curve fit. (F)

*Input values for a launch azimuth of 90 degrees.

Payload Determination for High Energy Missions

NXC3		(2)	(1) Number of points in SLOPC curve before critical C3 value. (I) (2) Number of points in SLOPC curve after critical C3 value. (I)
NORC3		(2)	(1) Order of interpolation for NXC3(1) (2) Order of interpolation for NXC3(2)
SLOPC	(C3)	(15)	Fraction of cut-off weight to parking orbit left after burning to the desired C3. The first NXC3(1) values are determined using a 100 n mi parking orbit. (F)
C3COFF			C3 value where parking orbit altitude changes from 100 n mi in order to increase payload at desired C3. (F)
NSOPAL			Number of points in the SLOPAL array (I)
NORPAL			Order of interpolation of SLOPAL. (I)
SLOPAL	(C3)	(5)	Dependent array for determining the optimum parking orbit altitude to get maximum payload to a C3 greater than C3COFF. (F)

Stage Impact Trace to Parking Orbit for High Energy Missions*

NXLLC3		(3)	(1) Number of coefficients for the latitude curve fit. Set equal to 7. (I) (2) Number of C3 values.
NORLC3		(3)	(1) Order of interpolation for NXLLC3(1). Set equal to 7. (I) (2) Order of interpolation for NXLLC3(2).
XLLC3	(Engine cut-off time, C3)	(15)	Independent variable array. Input 7 zeroes followed by NXLLC3(2) C3 values, (F)
F4	(Engine cut-off time, C3)	(70)	Coefficients of the impact point latitude versus engine cut-off time curve fit. (F)
F5	(Engine cut-off time, C3)	(70)	Coefficients of the impact point longitude versus engine cut-off time curve fit. (F)

Payload Determination to Circular Polar Orbits

NXYP1**		(2)	(1) Number of altitudes. (I) (2) Number of different times that yaw steering begins.
NORYP1		(2)	(1) Order of interpolation for NXYP1(1). (I) (2) Order of interpolation for NXYP1(2).

*Input values for a launch azimuth of 90 degrees.
**Refers to a launch azimuth of 45 degrees.

XYP1	(Altitude, Time of Yaw Steering initiation)	(8)	Independent variable array. First NXYP1(1) values of altitude followed by NXYP1(2) values of yaw times.
FY1	(Altitude, Time of Yaw Steering initiation)	(25)	Dependent array of cut-off weights to circular polar orbits. (F)
NXYP2 NORYP2 XYP2 FY11			Defined as above for a launch azimuth 145 degrees

Payload Determination to Elliptical Polar Orbits

NXEY1**			Number of points in the SOPEY1 array. (I)
NOEY1			Order of interpolation of SOPEY1. (I)
SOPEY1	(Perigee altitude)	(5)	Slopes of the payload versus apogee altitude lines as a function of perigee altitude. Perigee altitudes should correspond to the altitudes used in XYP1. (F)
NXEY2 NOEY2 SOPEY2			Defined as above but for a launch azimuth of 145 degrees.

Stage Impact TRACE to Circular Polar Orbits

NXYLP1**		(3)	(1) Number of coefficients stored for impact point trace as a function of engine cut-off time. Set equal 7. (I) (2) Number of altitudes (3) Number of times yaw steering begun.
NRYLP1		(3)	(1) Order of interpolation for NXYLP1(1). Set equal to 7. (I) (2) Order of interpolation for NXYLP1(2). (3) Order of interpolation for NXYLP1(3).
XYLLP1	(Time of engine cut-off, altitude, time yaw steering initiated	(15)	Independent variable array. Input 7 zeroes followed by NXYP1(2) values of altitude followed by NXYP2(3) values when yaw steering was begun. (F)
FY2	(Time of engine cut-off, altitude, time yaw steering initiated	(105)	Coefficients of the impact point latitude versus engine cut-off time curve fit. (F)
FY3	(Time of engine cut-off, altitude, time yaw steering initiated	(105)	Coefficients of the impact point longitude versus engine cut-off time curve fit. (F)
NXYLP2 NRYLP2 XYLLP1 FY12 FY13			Defined as above, but for a launch azimuth of 145 degrees.

**Refers to a launch azimuth of 45 degrees

Payload Determination for Synchronous Orbits

NXSY			Number of points in XSAZ array. (I)
NRSY			Order of interpolation in XSAZ. (I)
XSAZ	(Launch azimuth)	(10)	Independent array of launch azimuth for 100 n mi circular parking orbits. (F)
FY4	(Launch azimuth)	(10)	Dependent array of cut-off weights to 100 n mi circular parking orbits. (F)

Stage Impact TRACE to 100 n mi Parking Orbit for Synchronous Orbits

NLXY			Number of coefficients for the stage impact trace versus engine cut-off time curve fit. Set equal to 7. (I)
NLRSY			Order of interpolation. Set equal to 7. (I)
FY14	(Launch azimuth)	(7)	Coefficients of the impact point latitude versus engine cut-off time curve fit. For a launch azimuth of 90 degrees. (F)
FY15	(Launch azimuth	(7)	Coefficients of the impact point longitude versus engine cut-off time curve fit. For a launch azimuth of 90 degrees. (F)

Vehicle Information

AME		(4)	Name of vehicle etc. (A) 4 word description
NSTGS			Number of stages for the vehicle. (I)
VENOM	(Item, Stage) (J = 1, 3), (I = 1, NSTGS)	(3,3)	Nominal values of the item for each stage. (F) Item 1 Thrust (Pounds) Item 2 ISP (Seconds) Item 3 Stage Weight (Pounds)

Exchange Ratio Information as a Function of Altitude

THRUST

NPTA(I)***	(Stage)	(3)	(i) Number of altitude points for XETA for the i^{th} stage. (I)
NRTA(I)	(Stage)	(3)	(i) Order of interpolation for the i^{th} stage.
XETA(J,I)	(Altitude, Stage)	(15)	Particular values of altitude used with the thrust exchange ratios for each stage. (F)
FTA(J,I)	(Altitude, stage)	(15)	Values of the thrust exchange ratios for each altitude and stage. (F)

***I = 1, NSTGS
J = 1, No. of Altitude Points

ISP

NPISA(I)	(Stage)	(3)	(i) Number of altitude points for XEISA array for the i^{th} stage. (I)
NRISA(I)	(Stage)	(3)	(i) Order of interpolation for the i^{th} stage. (I)
XEISA(J,I)	(Altitude, stage)	(15)	Particular values of altitude used with the ISP exchange ratios for each stage. (F)
FISPA(J,I)	(Altitude, stage)	(15)	Values of the ISP exchange ratios for each altitude and stage. (F)

Stage Weight

NPSWA(I)	(Stage)	(3)	(i) Number of altitude points for XESWA array for the i^{th} stage. (I)
NRSWA(I)	(Stage)	(3)	(i) Order of interpolation for the i^{th} stage. (I)
XESWA(J,I)	(Altitude, stage)	(15)	Particular values of altitude used with the stage weight exchange ratios for each stage. (F)
FSWA(J,I)	(Altitude, stage)	(15)	Values of the stage weight exchange ratios for each altitude and stage. (F)

Exchange Ratio Information as a Function of C_3

FTC
XETC
NPTC
NRTC
FISPC
XEISC
NRISC
FSWC
XESWC
NPSWC
NRSWC

Defined as above but for C_3 instead of altitude.

Appendix C

INPUT DATA DESCRIPTION

C.1 GENERAL

This section contains the dictionary describing the program options, mission, and vehicle parameters. The following instructions should be read carefully before trying to input the program. Following the option dictionary is a chart describing the sections of the mission and vehicle parameters which must be input for the various options selected.

All data are input to the program through NAMELIST. NAMELIST allows the reading and writing of data without the use of an input/output list. Each NAMELIST must be referenced by name. The NAMELIST name for this program is "ONE". The first column of all data cards is ignored by NAMELIST. The first data card begins with a $ in column 2 followed by the NAMELIST name "ONE". The name ONE is followed by one or more blanks. Each item is input as variable name=constant followed by a comma. A $END following the last item indicates an end of data to the NAMELIST. The reader is referred to reference 9 for a complete description of NAMELIST.

There is a slight variation in the NAMELIST formats between computers. Input problems can be avoided by consulting a programmer.

C.2 INPUT DICTIONARY

C.2.1 Option Dictionary

The following is a description of the options available in the program and the FORTRAN variable associated by each. To select an option, set the appropriate variable name equal to number by the particular choice.

C.2.1.1 Primary Computer Objectives. The primary objective of the program is to determine payload to various orbits as a function of mission and vehicle parameters. The program is also capable of determining a stage impact trace as a function of mission parameters. The user must select one of the following options:

NOPT1 = 1 Payload determination as a function of mission parameters.
2 Stage impact trace as a function of mission parameters.
3 Payload determination as a function of vehicle parameters.

C.2.1.2 Types of Missions. The following missions are available in the program. The low earth orbits are achieved with no yaw steering. Polar orbits are achieved by using a launch azimuth of 45 or 145 degrees. The high energy and synchronous orbits proceed first through a 100 n mi circular parking orbit. The user must select one.

NOPT2 = 1 Low Earth Orbit
2 Polar Orbit
3 High Energy
4 Synchronous Orbit

C.2.1.3 Types of Orbits. There are three types of orbits available for low earth and polar orbits. The user must pick one if NOPT2 is equal to 1 or 2.

NOPT3 = 1 Circular
2 Elliptical
3 Perigee injection onto an elliptical orbit and circulariation at apogee..

The following options are initially set to 1 by the program. If the first choice of each option will satisfy the user's needs, then no further input is necessary from this section.

C.2.1.4 Payload Option. This option gives the user the choice of having the final answer printed as cutoff weight, gross or net payload.

NOPT4 = 1 Cutoff Weight (pounds)
2 Gross Weight (pounds)
3 New Weight (pounds)

C.2.1.5 Output of Option 1. This option is only used if NOPT2 and NOPT3 equal one. The payload output may be printed as a function of azimuth running across the page and altitude running down, or the reverse, where altitude runs across and azimuth down.

NOPT5 = 1 Payload output as a function of launch azimuth across the page with altitude down the page.

2 Payload as a function of altitude across page and azimuth down.

C.2.1.6 Output of Option 2. This option allows the user to make multiple copies of the output.

NOPT6 = N, Prints N copy(s) of output.

This option gives the user two print options. The output from each vehicle may be printed on separate pages, or as much information as possible for several vehicles may be printed on one page.

NOPT7 = 1 Prints results from 1 vehicle per page.

2 Prints results for several vehicles per page.

C.2.1.7 Curve Fit Option. This option allows the user to curve fit the stage impact traces. The latitude and longitude points are input as a function of time. The program then curve fits the points using the method of least squares. The coefficients are output on cards which can be stored in the data banks.

To select this option input the following card, starting in column 2. $ONE NOPT8 = 2 $END. This is done because the curve fit routines read pertinent information in themselves. They also use the NAMELIST name ONE. The curve fit routine is designed to generate data that can be stored directly in the data banks. In order that this option perform properly input the information to be curve fitted in the following manner. Indicate the number of altitudes or C3s for which impact traces are to be curve fitted. Do this by inputing the following card, starting in column 2. $ONE NSE = n $END. Here n is the number of altitudes or C3's. Then the routine reads the first set of latitude and time point, which correspond to the first attitude or C3. After curve fitting these points the coefficients are stored on cards. Then the program reads another set of latitude points and repeats the process. After the program has curve fitted the latitude points for all altitudes or C3, then input the longitude points in the same manner. For each curve fit, the data is input in NAMELIST $ONE as described in Section 8. For each complete set of latitude and longitude coefficients there are two additional cards punched.

The first card contains, starting in column 7
DATA FY--/

The following cards contain the coefficients. To use these cards, insert the correct number for this data in the -- locations, then insert these cards into the data bank for the appropriate vehicle. See Appendix B, Section 3 to determine the appropriate FY name.

C.3 PARAMETER OPTION COORDINATION

This section is to help the user determine which vehicle and mission parameters must be input for the particular choices of the options selected. Listed below are the various combinations of Options 1, 2, and 3 that the program will run. Under each combination is a list of the sections from Dictionary C.4 that must be input to obtain a run.

Section 1 of dictionary C.4 must always be input for any combination of option except for option 8.

NOPT1	NOPT2	NOPT3	SECTIONS
1	1	1, 2, 3	2
1	2	1, 2, 3	2, 4
1	3	1	2, 3
1	4	1	6
2	1	1, 2, 3	2, 5
2	2	1, 2, 3	2, 4, 5
2	3	1	2, 3, 5
2	4	1	5, 6
3	1	1	2, 7
3	3	1	2, 3, 7

NOPT8	8

C.4 MISSION AND VEHICLE PARAMETER INPUT DICTIONARY

The following dictionary is divided into sections which are separated by horizontal bars. If a particular section is applicable to the user's problem, then all information between the bars must be supplied. The variable type is also listed. An "F" indicates a floating point variable, an "I" indicates an integer.

Section	Variable Name	Maximum Dimension	Description
1	NOVEH	1	The number of different vehicles to be considered for the run. (I)
	NOVE(I)	10 (I=1, NOVEH)	Particular vehicles desired, see Vehicle Dictionary E. (I)
2	NALT	1	Number of different altitudes to be Considered. (I)
	ALT(I)	10	Particular altitudes for circular orbits. (F).
		(I=1, NALT)	Perigee altitudes for elliptical orbit.
	NFLAG		= 1 input launch azimuth
			= 2 input orbital inclination (I)
	IFF		Use only if NFLAG=2.
			= 1 northerly launch
			= 2 southernly launch
	NAZ	1	Number of different launch azimuths (I)
	AZ(I)	10 (I=1, NAZ)	Particular launch azimuth desired. Set to 45 or 145 for polar orbits. (F)
	AINC(I)	10 (I=1, NAZ)	Particular values of orbital inclination (F)
	NALTA	1	Number of apogee altitudes to consider. (I)
	ALTA(I)	10 (I=1, NALTA)	Particular apogee altitudes desired. (F)
3	NC3	1	Number of high energy missions desired. (I)
	C3(I)	10 (I=1, NC3)	Particular value of C3 desired. (F)
4	NTYAW(I)	10 (I=1, NOVEH)	Number of times to consider yaw steering to begin for the Ith vehicle. (I)
	TYAW(J,I)	(5, 10) (I=1, NOVEH) (J=1, 5)	Time after liftoff to begin yaw steering. (F)

Section	Variable Name	Maximum Dimension	Description
5	NTBOS(I)	10 (I=1, NOVEH)	Number of time points in stage impact trace per vehicle. (I)
	DTBOS(I)	10 (I=1, NOVEH)	Time step in impact trace per vehicle. (F)
	TBOST(I)	10 (I=1, NOVEH)	Time to start impact trace per vehicle. (SEC) (F)
6	NSYNC	1	= 1 Equatorial synchronous orbit hover point specified. (I) = 2 Equatorial synchronous orbit Launch azimuth for parking orbit specified. (Also Input Section 2) = 3 Synchronous orbit altitude hover point specified. = 4 Synchronous orbit altitude launch azimuth for parking orbit specified. (Also Input Section 2)
	ALOND(I)	10 (I=1, NOVEH)	Longitude (degrees) of desired hover point per vehicle. (F)

The following information is for obtaining runs for variations in the vehicle parameters. First the desired mission parameters should be selected, then the stages and items selected for change must be indicated to the program in increasing order. That is, if two stages of a three-stage vehicle are to changed, then the lower must be indicated first. The items must be taken in the following order: thrust, ISP, and stage weight.

Section	Variable Name	Maximum Dimension	Description
7	NSTAGS(I)	10 (I=1, NOVEH)	Number of stages on which parameter variations are to be performed per vehicle. (I)
	NSTAG(J,I)	(3,10) (J=1, NSTAGS(I)) (I=1, NOVEH)	Particular stage to be changed. (I)
	ITEMS (J,I)	(3,10) (J=1, NSTAGS(I)) (I=1, NOVEH)	Number of items (vehicle parameters) to be varied for a given stage and vehicle. (I)
	ITEM (K,L)	(3,3) (K=1, ITEMS (J,I)) (L=1, NSTAGE (J,I))	Particular item to be varied. (I) = 1 For a variation in Thrust = 2 For a variation in ISP = 3 For a variation in Stage weight

Section	Variable Name	Maximum Dimension	Description
	DELPR(K,L)	(3,3) (K=1, ITEMS(J,I)) (L=1, NSTAGE(J,I))	Desired change in a particular item per stage. (F) DELTA ITEM 1 in % DELTA ITEM 2 in sec. DELTA ITEM 3 in %
8	NSET		Number of different altitudes or C3 for which trace is desired. (I)
	NP		Number of points for the independent variable. (I)
	TIME	(100)	The independent variable time. (F)
	TRACE	(100)	Dependent variable (latitude or longitude) corresponding to time array. (F)

C.5 VEHICLE DICTIONARY

This dictionary defines the current vehicles and associated missions for which data are stored. To select a vehicle, set the variable NOVE(I) from Dictionary D equal to the correponding number.

Number	Vehicle	Mission	Exchange Ratio Function Altitude	Exchange Ratio Function C3
1	SATURN IB	1, 2, 3	*	
2	SATURN V 516	1, 3		*
3	SATURN V 2 stage	1, 2, 3	*	
4	TITAN 3C	1		
5	TITAN 3M	1		

Mission Legend:

1 Low Earth Orbit

2 Polar Orbit

3 High Energy

4 Synchronous Orbit

Appendix D

SAMPLE RUNS

This appendix contains sample runs from the program. These runs are to demonstrate the various output formats which are available.

MISSION - PAYLOAD DETERMINATION FOR LOW EARTH CIRCULAR ORBIT

VEHICLE - SA 212 PMR J2

LAUNCH AZIMUTH (DEGREES)	35.00	70.00	90.00	130.00
ORBITAL INCLINATION (DEGREES)	58.95	33.40	28.39	39.32
ALTITUDE (N-MI)	CUT OFF WEIGHT (POUNDS)			
100.00	65781.18	68732.00	69241.00	67457.70
200.00	58386.32	60843.00	61265.00	59785.36
270.00	51980.68	54028.84	54381.20	53146.79
550.00	24218.33(03)	24861.52(02)	24974.26(02)	24582.13(02)

(**) Estimate of extrapolation error - %

MISSIØN - STAGE IMPACT PØINT TRACE FØR LØW EARTH CIRCULAR ØRBIT

VEHICLE - SA 212 PWR J2

ALTITUDE (N-MI)
100.00

LAUNCH AZIMUTH (DEGREES)	45.00		70.00		90.00		130.00	
ØRBITAL INCLINATIØN (DEGREES)	49.47		33.40		28.39		39.32	
ENGINE CUT ØFF TIME (SECØNDS)	STAGE IMPACT PØINT (DEGREES)							
	LØNG	LAT	LØNG	LAT	LØNG	LAT	LØNG	LAT
50.00	-80.61	28.31	-80.63	28.34	-80.64	28.36	-80.62	28.40
100.00	-79.48	29.31	-79.13	28.82	-79.04	28.36	-79.40	27.50
150.00	-77.19	31.06	-76.22	29.56	-76.05	28.18	-77.25	25.69
200.00	-75.22	32.62	-73.70	30.24	-73.45	28.08	-75.37	24.16
250.00	-73.60	33.84	-71.67	30.76	-71.39	27.97	-73.90	22.93
300.00	-71.90	34.96	-69.63	31.15	-69.38	27.74	-72.54	21.65
350.00	-69.67	36.29	-67.05	31.55	-66.88	27.37	-70.91	20.00
400.00	-66.63	38.07	-63.56	32.10	-63.54	26.90	-68.71	17.83
450.00	-62.22	40.51	-58.66	32.80	-58.91	26.21	-65.65	14.83

MISSIØN - PAYLØAD DETERMINATIØN FØR LØW EARTH ELLIPTICAL ØRBIT

VEHICLE - SA 212 PMR J2

LAUNCH AZIMUTH (DEGREES)		45.00	70.00	90.00	130.00
ØRBITAL INCLINATIØN (DEGREES)		49.47	33.40	28.39	39.32
ALTITUDE (N-MI)		CUT ØFF WEIGHT (PØUNDS)			
PERIGEE	APØGEE				
100.00	400.00	64295.25	66151.25	66660.25	64876.95
100.00	500.00	60854.25	62710.25	63219.25	61435.95
200.00	400.00	57809.00	59353.00	59775.00	58295.36
200.00	500.00	57064.00	58608.00	59030.00	57550.36
270.00	400.00	51924.92	53212.42	53564.77	52330.36
270.00	500.00	51296.90	52584.40	52936.75	51702.34

MISSIØN - PAYLØAD DETERMINATIØN FØR HIGH ENERGY MISSIØNS

VEHICLE - SA 212 PMR J2

PARKING ØRBIT ALTITUDE (N-MI)	C3 (KM/SEC)**2	LAUNCH AZIMUTH (DEGREES) 45.00 / ØRBITAL INCLINATIØN (DEGREES) 49.47	70.00 / 33.40	90.00 / 28.39	130.00 / 39.32
		CUT ØFF WEIGHT (PØUNDS)			
100.00	-50.00	51251.53	52673.91	53063.99	51697.33
100.00	-40.00	39456.84	40551.88	40852.19	39800.05
100.00	-20.00	21935.33	22544.10	22711.05	22126.13

MISSIØN - EQUITØRIAL SYNCHRØNØUS ØRBIT

VEHICLE - ENTERPRISE

LØNGITUDE ØF ARRIVAL 210.00 DEGREES

DEPARTURE FRØM THE ASCENDING NØDE

LAUNCH AZIMUTH (DEGREES)	113.99	98.29	86.95	73.55
NUMBER ØF PASSES THRØUGH NØDE	2	3	4	5
TIME FRØM ØRBIT INJECTIØN TØ FIRST DESCENDING NØDE (SEC)	470.38	795.97	1077.17	1391.57
TIME FRØM FIRST DESCENDING NØDAL PASSAGE TØ DESIRED DEPARTURE NØDE (SEC)	7932.00	13220.00	18508.00	23796.00

LAUNCH AZIMUTH (DEGREES)	113.99	98.29	86.95	73.55
CUT ØFF WEIGHT TØ ØRBIT (PØUNDS)	96703.21	99502.93	99390.91	96727.55

This vehicle and the payload information are fictitious, they are for display purposes only.

MISSIØN-PAYLØAD DETERMINATIØN FØR LØW EARTH CIRCULAR ØRBIT
ALTITUDE 100.00 (N-MI)
LAUNCH AZIMUTH 45.00 (DEGREES) ØRBITAL INCLINATIØN 49.47 (DEGREES)
VEHICLE SA 212 PMR J2

*** EXCHANGE RATIØ ANALYSIS ***

STAGE	PARAMETER	NØMINAL VALUE	PERTURBATIØN	PAYLØAD EXCHANGE RATIØ	PAYLØAD PERTURBATIØN
1					
	THRUST	205000.00 LB	5.00 PERCENT	305.94 LB/PERCENT	1530. LB
	ISP	263.40 SEC	5.00 SEC	250.75 LB/SEC	1254. LB
	STAGE WT	106124.00 LB	1000.00 LB	-0.14 LB/LB	-139. LB
2					
	THRUST	190000.00 LB	2.00 PERCENT	201.00 LB/PERCENT	402. LB
	ISP	428.47 SEC	10.00 SEC	231.60 LB/SEC	2316. LB
	STAGE WT	24460.00 LB	500.00 LB	-1.00 LB/LB	-500. LB

				TØTAL	4862. LB
				CUT ØFF WEIGHT	717[illegible]. LB

D-7

NORTHROP
HUNTSVILLE

TR-796-9-594

Appendix E

PROGRAM LISTING

```
$JOB           NORT-KILPA-BIN 215 ,491240,00,11,14CE
$EXECUTE       IBJOB
$IBJOB         MAP
$IBFTC MAIN
C      ********** APT PROGRAM **********
C
       COMMON /ALACOM/ ALA(500),ALO(500)
       COMMON /BOSCOM/ TBOST(10),DTBOS(10),NTBOS(10)
       COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
      1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
       COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
       COMMON /ECCOM/ SLOPE(5),NXE,NORE,SOPEY1(5),NXEY1,NOEY1
      1,SOPEY2(5),NXEY2,NOEY2
       COMMON /ERRCOM/ ERR(400),NERR,PAERR
       COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
      1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
      2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
       COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
      1,DELPR(3,3)
       COMMON /LINCOM/ LC,LMAX,NPAGE
       COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
      1,AINC(10),NOVEH,NOVE(10)
       COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
       COMMON /PLCOM/ PL(10),XX(4)
       COMMON /RATCOM/ FTA(15),XETA(15),NPTA(3),NRTA(3)
      1,FTC(15),XETC(15),NPTC(3),NRTC(3)
      2,FISPA(15),XEISA(15),NPISA(3),NRISA(3)
      3,FISPC(15),XEISC(15),NPISC(3),NRISC(3)
      4,FSWA(15),XESWA(15),NPSWA(3),NRSWA(3)
      5,FSWC(15),XESWC(15),NPSWC(3),NRSWC(3)
       COMMON /ROTCOM/ T90(3,3),T90I(3,3),TAZ(3,3),ABC(3,3)
       COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
      1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
       COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
       COMMON /WHSCOM/ MSTG(3),MITH(3,3)
       COMMON /WGTCOM/ GPFC,ANETP
       COMMON /XPCOM/ X(10),NX(2),NORX(2),XLLP(16),NXLLP(3),NORLLP(3)
      1,XLLC3(15),NXLLC3(3),NORLC3(3)
       COMMON /XYWCOM/ XYP1(8),NXYP1(2),NORYP1(2)
      1,XYP2(8),NXYP2(2),NORYP2(2)
      2,XYLLP1(15),NXYLP1(3),NRYLP1(3)
      3,XYLLP2(15),NXYLP2(3),NRYLP2(3)
       COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
C
       CALL INFO
     1 CONTINUE
       CALL INPT1 (1)
     2 CONTINUE
       GO TO (3,6,9),NOPT1
C      ********** PAYLOAD DETERMINED **********
     3 CONTINUE
       IF(NOPT3.EQ.5) GO TO 10
       GO TO (4,5),NOPT2
     4 CONTINUE
C      ********** NO YAW STEERING **********
```

```
      CALL PLNYW
      GO TO 11
    5 CONTINUE
C     ********** YAW STEERING **********
      CALL PLYAW
      GO TO 11
    6 CONTINUE
C     ********** IMPACT POINT TRACE DETERMINED **********
      GO TO (7,8),NOPT2
    7 CONTINUE
C     ********** NO YAW STEERING **********
      CALL LLNYW
      GO TO 11
    8 CONTINUE
C     ********** YAW STEERING **********
      CALL LLYAW
      GO TO 11
    9 CONTINUE
C     ********** EXCHANGE RATIO **********
      CALL EXRAT
      GO TO 11
   10 CONTINUE
C     ********** SYNCHRONOUS ORBIT **********
      CALL SYNCH
   11 CONTINUE
      CALL INPT1 (2)
      GO TO 2
      END

$IBFTC SUB10
      BLOCK DATA
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /ROTCOM/ T90(3,3),T90I(3,3),TAZ(3,3),ABC(3,3)
      DATA U,G,RE,CNMK,PIE,RAD/398603.2,9.8E-03,6378.165,1.852,3.14159,
     157.2957795/
      DATA T90 /.1443107,.986461,-.7790164E-01,
     1-.8680595,.1639947,.4685907,.4750221,.5662706E-06,.8799736/
      DATA T90I /.1443107,-.8680595,.4750221,
     1.986461,.1639947,.5662706E-06,-.7790164E-01,.4685907,.8799736/
      DATA NRIT2 /10/
      DATA NRED/5/,NRIT/6/
      FND

$IBFTC SUB11
      SUBROUTINE INPT1 (NNII)
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /BOSCOM/ TBOST(10),DTBOS(10),NTBOS(10)
      COMMON/LINCOM/ LC,LMAX,NPAGE
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
```

```
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
     1,DELPR(3,3)
      NAME LIST /ONE/ NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7,NOPT8
     1,NOVEH,NOVE,ALT,NALT,AZ,NAZ,NFLAG,IFF,ALTA,NALTA,C3,NC3
     2,TBOST,DTBOS,NTBOS,TYAW,NTYAW,AINC,ALOND,NSYNC
     3,NSTAGS,NSTAG,ITEMS,ITEM,DELPR
C     NOPT4 = 1  CUT OFF  WT
C     NOPT4 = 2  GROSS WT
C     NOPT4 = 3  NET WT
C     NC3S NUMBER OF C3 CURVES
C     NOVEH NUMBER OF VEHICLES
C     NOVE(I)PARTICULAR VEHICLE
C     IFF=0 INPUT AZ  IFF=1 INPUT INC
C     NFLAG=1 NORTH
C     NFLAG=2 SOUTH
      IF (NNII.GT.1) GO TO 1
      LMAX=54
      NOPT4=1
      NOPT5=1
      NOPT6=1
      NOPT7=1
      NOPT8=1
      IFF=0
      NFLAG=1
    1 CONTINUE
      RAD=57.2957795
      LC=0
      NPAGE=0
      READ (NRED,ONE)
      IF(NOPT8 .GT.1) GO TO 3
      IF (NOPT2.GT.2) NOPT3=NOPT2+1
      IF (NOPT2.GT.2) NOPT2=1
      IF(NOPT1.EQ.1.AND.NOPT3.EQ.2) NOPT5=1
      IF(NOPT1.EQ.1.AND.NOPT3.EQ.3) NOPT5=1
      IF(NOPT3.EQ.5) GO TO 2
      CALL INCAZ (NFLAG,IFF)
      WRITE (NRIT,ONE)
      RETURN
    2 CONTINUE
      ALT(1)=100.
      NALT=1
      NOPT5=1
      IF (NSYNC.EQ.2.OR.NSYNC.EQ.4) CALL INCAZ (1,0)
      WRITE(NRIT,ONE)


      RETURN
    3 CONTINUE
      CALL CURVE
      GO TO 1
      END
```

```
$IBFTC SUB12
      SUBROUTINE BANK(IG)
      GO TO (1,2,3,4,5,6,7,8),IG
    1 CONTINUE
      CALL VEH1
      RETURN
    2 CONTINUE
      CALL VEH2
      RETURN
    3 CONTINUE
      CALL VEH3
      RETURN
    4 CONTINUE
      CALL VEH4
      RETURN
    5 CONTINUE
      CALL VEH5
      RETURN
    6 CONTINUE
      RETURN
    7 CONTINUE
      RETURN
    8 CONTINUE
      RETURN
      END


$IBFTC SUB13
      SUBROUTINE PLNYW
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
     1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
     2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
      COMMON /XPCOM/ X(10),NX(2),NORX(2),XLLP(16),NXLLP(3),NORLLP(3)
     1,XLLC3(15),NXLLC3(3),NORLC3(3)
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
     1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
      COMMON /ECCOM/ SLOPE(5),NXE,NORE,SOPEY1(5),NXEY1,NOEY1
     1,SOPEY2(5),NXEY2,NOEY2
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON/WGTCOM/ GPFC,ANETP
  100 FORMAT(7E15.8)
  200 FORMAT(/9(1XE12.5))
      IF (NOPT1.NE.3) REWIND NRIT2
      DO 50 I=1,NOVEH
      NERR=0
      IF (NOPT1.EQ.3) GO TO 5
      CALL BANK(NOVE(I))
      WRITE(NRIT2) AME1,AME2,AME3,AME4
      WRITE(NRIT2) GPFC,ANETP
```

```
    5 CONTINUE
      DO 40 J=1,NAZ
      XX(1)=AZ(J)
      GO TO (10,10,10,20,30),NOPT3
   10 CONTINUE
C      CIRCULAR
      DO 12 K=1,NALT
      XX(2)=ALT(K)
      CALL INTLA (0,2,F1,X,NX,NORX,C,XX,PL(K))
      CALL ERROR(2,F1,X,NX,NORX,XX,ERR(1,K),1,2)
      IF (NOPT3.EQ.1) GO TO 12
      RP=ALT(K)*CNMK+RE
C      ELLIPSE
      DO 11 L=1,NALTA
      RA=ALTA(L)*CNMK+RE
      NSUB=NX(1)+1
      CALL INTLA (0,1,SLOPE,X(NSUB),NX (2),NORX(2),C,ALT(K),AM)
      PL(L)=PL(K)+AM*(ALTA(L)-ALT(K))
      IF (NOPT3.EQ.2) GO TO 11
C     ELLIPSE AND CIRCULARIZE
      DV=SQRT(U/RA)*(1.-SQRT(2.*RP/(RA+RP)))
      EX=EXP(-DV/(VENOM(2,NSTGS)*G))
      PL(L)=PL(L)*EX
   11 CONTINUE
      IF (NOPT1.EQ.3) GO TO 12
C
      WRITE (NRIT ,200) (ALTA(L),L=1,NALTA)

      WRITE (NRIT ,200) (PL(L),L=1,NALTA)
C
      WRITE(NRIT2) (PL(L),(ERR(LL,K),LL=1,4),L=1,NALTA)
   12 CONTINUE
      IF ((NOPT1.EQ.3).OR.(NOPT3.NE.1)) GO TO 40
C
      WRITE (NRIT ,200) (ALT(K),K=1,NALT)
      WRITE (NRIT ,200) (PL(K),K=1,NALT)
C
      WRITE(NRIT2) (PL(K),(ERR(LL,K),LL=1,4),K=1,NALT)
      GO TO 40
   20 CONTINUE
C      C3
      DO 25 K=1,NC3
      JC3=1
      IC3=1
      XX(2)=100.
      IF (C3(K).LE.C3COFF) GO TO 21
      CALL INTLA(0,1,SLOPAL,X,NSOPAL,NORPAL,C,C3(K),AM)
      XX(2)=AM*(C3(K)-C3COFF)
      JC3=NXC3(1)+1
      IC3=2
   21 CONTINUE
      CALL INTLA (0,2,F1,X,NX,NORX,C,XX,PL(K))
      CALL ERROR(2,F1,X,NX,NORX,XX,ERR(1,1),1,2)
```

```
      CALL INTLA (0,1,SLOPC(JC3),XC3(JC3),NXC3(IC3),NORC3(IC3),C
     1,C3(K),AM)
      PL(K)=PL(K)*AM
      IF (NOPT1.EQ.3) GO TO 25
C
      WRITE (NRIT ,200) XX(2),PL(K)
C
      WRITE(NRIT2) XX(2),PL(K),(ERR(LL,1),LL=1,4)
   25 CONTINUE
      GO TO 40
   30 CONTINUE
      PL(1)=0.
   40 CONTINUE
      IF(NOPT1 .EQ.3) RETURN
      WRITE (NRIT2) NERR
   50 CONTINUE
      DO 70 ITY=1,NOPT6
      CALL PRPLNY
   70 CONTINUE
      RETURN
      END

$IBFTC SUB14
      SUBROUTINE PLYAW
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
     1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
     2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /XYWCOM/ XYP1(8),NXYP1(2),NORYP1(2)
     1,XYP2(8),NXYP2(2),NORYP2(2)
     2,XYLLP1(15),NXYLP1(3),NRYLP1(3)
     3,XYLLP2(15),NXYLP2(3),NRYLP2(3)
      COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
     1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
      COMMON /ECCOM/ SLOPE(5),NXE,NORE,SOPEY1(5),NXEY1,NOEY1
     1,SOPEY2(5),NXEY2,NOEY2
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON/WGTCOM/ GPFC,ANETP
  100 FORMAT(7E15.8)
  200 FORMAT(/9(1XE12.5))
      REWIND NRIT2
      LAZ=1
      IF (AZ(1).GT.45.) LAZ=2
      DO 50 I=1,NOVEH
      NERR=0
      CALL BANK(NOVE(I))
      WRITE(NRIT2) AME1,AME2,AME3,AME4
      WRITE(NRIT2) GPFC,ANETP
```

```
      NYAW=NTYAW(I)
      DO 40 J=1,NYAW
      XX(2)=TYAW(J,I)
      GO TO (10,10,10,20,30),NOPT3
   10 CONTINUE
C     CIRCULAR ORBITS
      DO 16 K=1,NALT
      XX(1)=ALT(K)
      IF (LAZ.EQ.2) GO TO 11
      CALL INTLA (0,2,FY1,XYP1,NXYP1,NORYP1,C,XX,PL(K))
      CALL ERROR(2,FY1,XYP1,NXYP1,NORYP1,XX,ERR(1,K),2,2)
      GO TO 12
   11 CONTINUE
      CALL INTLA (0,2,FY11,XYP2,NXYP2,NORYP2,C,XX,PL(K))
   12 CONTINUE
      IF (NOPT3.EQ.1) GO TO 16
C     ********** ELLIPTICAL ORBIT **********
      RP=ALT(K)*CNMK+RE
      DO 15 L=1,NALTA
      IF (LAZ.EQ.2) GO TO 13
      NSUB=NXYP1(1)+1
      CALL INTLA (0,1,SOPEY1,XYP1(NSUB),NXEY1,NOEY1,C,ALTA(K),AM)
      GO TO 14
   13 CONTINUE
      NSUB=NXYP2(1)+1
      CALL INTLA (0,1,SOPEY2,XYP2(NSUB),NXEY2,NOEY2,C,ALTA(K),AM)
   14 CONTINUE
      PL(L)=PL(K)+AM*(ALTA(L)-ALT(K))
      IF (NOPT3.NE.3) GO TO 15
C     ********** ELLIPTICAL ORBIT AND CIRCULARIZE **********
      RA=ALTA(L)*CNMK+RE
      DV=SQRT(U/RA)*(1.-SQRT(2.*RP/(RA+RP)))
      EX=EXP(-DV/(VENOM(2,NSTGS)*G))
      PL(L)=PL(L)*EX
   15 CONTINUE
C
      WRITE (NRIT,200) (ALTA(L),L=1,NALTA)
      WRITE (NRIT,200) (PL(L),L=1,NALTA)
C
      WRITE(NRIT2) (PL(L),(ERR(LL,K),LL=1,4),L=1,NALTA)
   16 CONTINUE
      IF (NOPT3.GT.1) GO TO 40
C
      WRITE (NRIT,200) (ALT(K),K=1,NALT)
      WRITE (NRIT,200) ( PL(K),K=1,NALT)
C
      WRITE(NRIT2) (PL(K),(ERR(LL,K),LL=1,4),K=1,NALT)
      GO TO 40
   20 CONTINUE
C     ********** C3 ORBIT **********
   30 CONTINUE
   40 CONTINUE
      WRITE (NRIT2) NERR
```

```
   50 CONTINUE
      DO 70 ITY=1,NOPT6
      CALL PRPLY
   70 CONTINUE
      RETURN
      END


$IBFTC SUB15
      SUBROUTINE SYNCH
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
     1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
     2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON/WGTCOM/ GPFC,ANETP
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
      DATA CFTS/.37/,CFSA/.39/
  100 FORMAT(10(1XI3))
  200 FORMAT(9(1XE12.5))
C
C     CFTS THEORETICAL MASS FRACT FR 100 PO TO SYNCH L-AZ 90
C     CFSA THEORETICAL MASS FRACT FR 100 PO TO SYNCH ALT L-AZ 90
C     DVSYN MASS FRACTION FROM 100 N-MI PO TO SYNCH ORBIT L-AZ 90
C     DVSYA MASS FRACTION FROM 100 N-MI PO TO SYNCH ORBIT ALT L-AZ 90
C
      REWIND NRIT2
      CI1=.99905
      IF (NSYNC.GT.2) CI1=1.
      IF (NSYNC.LT.3) CF=(CFTS-DVSYN)/DVSYN
      IF (NSYNC.GT.2) CF=(CFSA-DVSYA)/DVSYA
      CF=1.-CF
      DO 40 I=1,NOVEH
      NERR=0
      CALL BANK(NOVE(I))
      EXPX=G*VENOM(2,NSTGS)
      WRITE (NRIT2) AME1,AME2,AME3,AME4
      WRITE(NRIT2) GPFC,ANETP
      GO TO (10,20,10,20),NSYNC
   10 CONTINUE
C
C     LONGITUDE OF HOVER POINT SPECIFIED
C
      CALL AZDET (I)
      CALL INCAZ (1,0)
C
      WRITE (NRIT2) JOLO,NOLO,NAZ,(AZ(IJK),IJK=1,NAZ)
      IF (NSYNC.EQ.3) WRITE (NRIT2) (AINC(IJK),IJK=1,NAZ)
```

```
C
      WRITE (NRIT,100) JOLO,NOLO,NAZ
      WRITE (NRIT,200) ALOND(I)
      WRITE (NRIT,200) (AINC(IK),IK=1,NAZ)
      WRITE (NRIT,200) (   AZ(IK),IK=1,NAZ)
C
   20 CONTINUE
      DO 30 J=1,NAZ
C
C     FY4 PL TO 100 N-MI PO FOR SYNCH ORBITS
C
      CALL INTLA (0,1,FY4,XSAZ,NXSY,NRSY,C,AZ(J),PL(J))
      CALL ERROR(1,FY4,XSAZ,NXSY,NRSY,AZ(J),ERR(1,J),1,1)
      CI2=COS((AINC(J)-2.5)/57.2957795)
      IF (NSYNC.GT.2) CI2=1.
      PL(J)=PL(J)*CF*EXP(-(SQRT(165.834-159.788*CI1)+
     1SQRT(12.002-9.814*CI2))/EXPX)
   30 CONTINUE
C
      WRITE(NRIT2) (PL(IK),(ERR(LL,IK),LL=1,4),IK=1,NAZ)
C
      WRITE (NRIT,200) (PL(IK),IK=1,NAZ)
C
      WRITE (NRIT2) NERR
   40 CONTINUE
      DO 70 ITY=1,NOPT6
      CALL SYNCOT
   70 CONTINUE
      RETURN
      END


$IBFTC SUB16
      SUBROUTINE EXRAT
      COMMON /OPTCOM/ NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON/MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /RATCOM/ FTA(15),XETA(15),NPTA(3),NRTA(3)
     1,FTC(15),XETC(15),NPTC(3),NRTC(3)
     2,FISPA(15),XEISA(15),NPISA(3),NRISA(3)
     3,FISPC(15),XEISC(15),NPISC(3),NRISC(3)
     4,FSWA(15),XESWA(15),NPSWA(3),NRSWA(3)
     5,FSWC(15),XFSWC(15),NPSWC(3),NRSWC(3)
      COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
     1,DELPR(3,3)
      COMMON/WGTCOM/ GPFC,ANETP
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
C
C     NSTGS                    NO STAGES
C     NSTAGS(VEH)              NO OF STAGES FOR PARAM VERATION
C     NSTAG (STAGE,VEH)        PARTICULAR STAGE
C     ITEMS(STAGE,VEH)         NO OF ITEMS TO CHANGE
C     ITEM (ITEM,STAGE)        PARTICULAR ITEM
```

```
C       ITEM 1 THRUST
C       ITEM 2 ISP
C       ITEM 3 STAGE WT
C       VENOM (ITEM,STAGE)       NOMINAL VALUES OF ITEMS
C       FIA   (ALT)              VALUES OF EXRATS FOR ITEM I
C       XEIA  (ALT)              ALTITUDES
C       NPIA   (STAGE)           NO OF POINTS
C       NRIA   (STAGE)           ORDER OF INTERPOIATION
C       FIC,XEIC,NPIC,NRIC   DEFINED AS ABOVE BUT FCNS OF C3
C
        REWIND NRIT2
        DO 60 I=1,NOVEH
        CALL BANK(NOVE(I))
        WRITE (NRIT2) NSTGS
        WRITE (NRIT2)  AME1, AME2, AME3,AME4
        WRITE(NRIT2) GPFC,ANETP
        WRITE (NRIT2) ((VENOM(JI,KS),JI=1,3),KS=1,NSTGS)
        CALL PLNYW
C
        WRITE (NRIT,200) PL(1)
C
        JSTG=NSTAGS(I)
        DPAYS=0.
        DO 50 J=1,JSTG
        KSTG=NSTAG(J,I)
        KITMS=ITEMS(KSTG,I)
        DPAYI=0.
        DO 40 K=1,KITMS
        LITM=ITEM (K,KSTG)
        GO TO (10,10,10,20,10),NOPT3
   10   CONTINUE
C       EX RAT FCN(ALT)
        GO TO (11,12,13),LITM
   11   CONTINUE
C       THRUST
        CALL ANUMB (KSTG,NPTA,NSUB)
        CALL INTLA (0,1,FTA(NSUB),XETA(NSUB),NPTA(KSTG),NRTA(KSTG),C
      1,ALT(1),ANS)
        GO TO 30
   12   CONTINUE
C       ISP
        CALL ANUMB (KSTG,NPISA,NSUB)
        CALL INTLA (0,1,FISPA(NSUB),XEISA(NSUB),NPISA(KSTG),NRISA(KSTG)
      1,C,ALT(1),ANS)
        GO TO 30
   13   CONTINUE
C       STAGE WT
        CALL ANUMB (KSTG,NPSWA,NSUB)
        CALL INTLA (0,1,FSWA(NSUB),XESWA(NSUB),NPSWA(KSTG),NRSWA(KSTG)
      1,C,ALT(1),ANS)
        GO TO 30
   20   CONTINUE
C       EX RAT FCN(C3)
        GO TO (21,22,23),LITM
```

```
   21 CONTINUE
C     THRUST
      CALL ANUMB (KSTG,NPTC,NSUB)
      CALL INTLA (0,1,FTC(NSUB),XETC(NSUB),NPTC(KSTG),NRTC(KSTG),C
     1,C3(1),ANS)
      GO TO 30
   22 CONTINUE
C     ISP
      CALL ANUMB (KSTG,NPISC,NSUB)
      CALL INTLA (0,1,FISPC(NSUB),XEISC(NSUB),NPISC(KSTG),NRISC(KSTG)
     1,C,C3(1),ANS)
      GO TO 30
   23 CONTINUE
C     STAGE WT
      CALL ANUMB (KSTG,NPSWC,NSUB)
      CALL INTLA (0,1,FSWC(NSUB),XESWC(NSUB),NPSWC(KSTG),NRSWC(KSTG)
     1,C,C3(1),ANS)
   30 CONTINUE
      PAY=DELPR(LITM,KSTG)*ANS
C
  200 FORMAT(7(1XE12.5))
      WRITE (NRIT,200) ANS,PAY
      WRITE (NRIT2) ANS,PAY
C
      DPAYI=DPAYI+PAY
   40 CONTINUE
      DPAYS=DPAYS+DPAYI
   50 CONTINUE
C
      PT=PL(1)+DPAYS
      WRITE (NRIT2) DPAYS,PT,(ERR(II,1),II=1,4),NERR
C
   60 CONTINUE
      CALL EXOUT
      RETURN
      END

$IBFTC SUB17
      SUBROUTINE LLNYW
      DIMENSION XRAY(3)
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
     1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
     2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /XPCOM/ X(10),NX(2),NORX(2),XLLP(16),NXLLP(3),NORLLP(3)
     1,XLLC3(15),NXLLC3(3),NORLC3(3)
      COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
     1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
      COMMON /ECCOM/ SLOPE(5),NXE,NORE,SOPEY1(5),NXEY1,NOEY1
     1,SOPEY2(5),NXEY2,NOEY2
      COMMON /BOSCOM/ TBOST(10),DTBOS(10),NTBOS(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
```

```
      COMMON /ROTCOM/ T90(3,3),T90I(3,3),TAZ(3,3),ABC(3,3)
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON /ALACOM/ ALA(500),ALO(500)
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
  100 FORMAT(7E15.8)
  200 FORMAT(/9(1XF12.5))
      REWIND NRIT2
      NDUM=NALT
      IF(NOPT3.FQ.4)NDUM=NC3
      DO 40 I=1,NOVEH
      CALL BANK(NOVE(I))
      WRITE(NRIT2) AME1,AME2,AME3,AME4
      NBOS=NTBOS(I)
      DT=DTBOS(I)
      DO 30 J=1,NAZ
      CALL MATAZ (AZ(J))
      DO 20 K=1,NDUM
      XX(1)=TBOST(I)-DT
      XX(2)=ALT(K)
      IF(NOPT3.EQ.4)XX(2)=C3(K)
      IF(NOPT3.EQ.5) XX(2)=AZ(J)
      DO 10 L=1,NBOS
      XX(1)=XX(1)+DT
      GO TO (1,1,1,2,3) ,NOPT3
    1 CONTINUE
C     CIRCULAR ORBIT
C     ELLIPTICAL ORBIT
      CALL INTLA (1,2,F2,XLLP,NXLLP,NORLLP,C,XX,ALAT)
      CALL INTLA (1,2,F3,XLLP,NXLLP,NORLLP,C,XX,ALON)
      GO TO 5
    2 CONTINUE
C     C3 ORBIT
      CALL INTLA (1,2,F4,XLLC3,NXLLC3,NORLC3,C,XX,ALAT)
      CALL INTLA (1,2,F5,XLLC3,NXLLC3,NORLC3,C,XX,ALON)
      GO TO 5
    3 CONTINUE
C     SYNCHRONOUS ORBIT
      CALL INTLA(1,1,FY14,XLSAZ,NXLSY,NLRSY,C,XX,ALAT)
      CALL INTLA(1,1,FY15,XLSAZ,NXLSY,NLRSY,C,XX,ALON)
    5 CONTINUE
      XRAY(1)=COS(ALON/RAD)*COS(ALAT/RAD)
      XRAY(2)=SIN(ALON/RAD)*COS(ALAT/RAD)
      XRAY(3)=SIN(ALAT/RAD)
      CALL MAT (TAZ,XRAY,XRAY,3,3,1)
      ALO(L)=RAD*ATAN2(XRAY(2),XRAY(1))
      ALA(L)=RAD*ATAN2(XRAY(3),SQRT(XRAY(1)**2+XRAY(2)**2))
   10 CONTINUE
C
      WRITE (NRIT,200) (ALA(L),L=1,NBOS)
      WRITE (NRIT,200) (ALO(L),L=1,NBOS)
C
      WRITE (NRIT2) (ALA(L),L=1,NBOS)
      WRITE (NRIT2) (ALO(L),L=1,NBOS)
   20 CONTINUE
```

```
   30 CONTINUE
   40 CONTINUE
      DO 70 ITY=1,NOPT6
      CALL PRLLY
   70 CONTINUE
      RETURN
      END

$IBFTC SUB18
      SUBROUTINE LLYAW
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
     1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
     2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /XYWCOM/ XYP1(8),NXYP1(2),NORYP1(2)
     1,XYP2(8),NXYP2(2),NORYP2(2)
     2,XYLLP1(15),NXYLP1(3),NRYLP1(3)
     3,XYLLP2(15),NXYLP2(3),NRYLP2(3)
      COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
     1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /BOSCOM/ TBOST(10),DTBOS(10),NTBOS(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /PLCOM/ PL(10),XX(4)
      COMMON /ALACOM/ ALA(500),ALO(500)
  100 FORMAT(7E15.8)
  200 FORMAT(/9(1XE12.5))
      REWIND NRIT2
      LAZ=1
      IF (AZ(1).GT.45.) LAZ=2
      DO 50 I=1,NOVEH
      CALL BANK(NOVE(I))
      WRITE(NRIT2) AME1,AME2,AME3,AME4
      NYAW=NTYAW(I)
      NBOS=NTBOS(I)
      DT=DTBOS(I)
      XX(1)=TBOST(I)-DT
      DO 40 J=1,NYAW
      XX(3)= TYAW(J,I)
      GO TO (10,10,10,20,30),NOPT3
   10 CONTINUE
C     C  E  EC
      DO 14 K=1,NALT
      XX(1)=TBOST(I)-DT
      XX(2)= ALT(K)
      DO 13 L=1,NBOS
      XX(1)=XX(1)+DT
      IF (LAZ.EQ.2) GO TO 11
      CALL INTLA (1,3,FY2,XYLLP1,NXYLP1,NRYLP1,C,XX,ALA(L))
      CALL INTLA (1,3,FY3,XYLLP1,NXYLP1,NRYLP1,C,XX,ALO(L))
      GO TO 12
```

```
   11 CONTINUE
      CALL INTLA (1,3,FY12,XYLLP2,NXYLP2,NRYLP2,C,XX,ALA(L))
      CALL INTLA (1,3,FY13,XYLLP2,NXYLP2,NRYLP2,C,XX,ALO(L))
   12 CONTINUE
   13 CONTINUE
C
      WRITE (NRIT,200) (ALA(L),L=1,NBOS)
      WRITE (NRIT,200) (ALO(L),L=1,NBOS)
C
      WRITE (NRIT2) (ALA(L),L=1,NBOS)
      WRITE (NRIT2) (ALO(L),L=1,NBOS)
   14 CONTINUE
      GO TO 40
C     C3
   20 CONTINUE
   30 CONTINUE
   40 CONTINUE
   50 CONTINUE
      DO 70 ITY=1,NOPT6
      CALL PRLLY
   70 CONTINUE
      RETURN
      END


$IBFTC SUB19
      SUBROUTINE PRPLNY
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /ERRCOM/ ERR(4,10,10),NERR,PAERR
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      COMMON /ALACOM/ PL(50,10),POA(500)
      COMMON/WGTCOM/ GPFC,ANETP
  100 FORMAT(7E15.8)
  200 FORMAT(/14XF7.2,11X,5(F10.2,4A1,1X))
  300 FORMAT(/9XF7.2,4XF7.2,5X,5(F10.2,4A1,1X))
  400 FORMAT(/9XF7.2,3XF7.2,6X5(F10.2,4A1,1X))
  500 FORMAT(/4XF7.2,10XF7.2,4X5(F10.2,4A1,1X))
 3000 FORMAT(//,13X,37H* (XX) EXTRAPOLATION BEYOND DATA BANK)
      REWIND NRIT2
      DO 60 I=1,NOVEH
      READ(NRIT2) AME1,AME2,AME3,AME4
      READ(NRIT2) GPFC,ANETP
      IF(NOPT7.EQ.1.OR.I.EQ.1) CALL MISS
      CALL CAPT
      GO TO (1,10,10,20,30),NOPT3
    1 CONTINUE
C     CIRCULAR ORBIT
      DO 2 K=1,NAZ
      READ(NRIT2) (PL(K,L),(ERR(LL,K,L),LL=1,4),L=1,NALT)
```

```
    2 CONTINUE
      IF(NOPT4.GT.1) CALL REDUCR(PL,NAZ,NALT)
      CALL OTPNY(I,POA)
      GO TO (3,5),NOPT5
    3 CONTINUE
C     AZ ACROSS
      DO 4 K=1,NALT
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,200) ALT(K),(PL(L,K),(ERR(LL,L,K),LL=1,4),L=1,NAZ)
    4 CONTINUE
      GO TO 50
C       AZ  DOWN
    5 CONTINUE
      DO 6 K=1,NAZ
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,300)AZ(K),AINC(K),(PL(K,L),(ERR(LL,K,L),LL=1,4),L=1,NALT)
    6 CONTINUE
      GO TO 50
   10 CONTINUE
C     ELLIPTICAL ORBITS
      DO 11 J=1,NAZ
      MU1=0
      MU2=NALTA
      DO 331 K=1,NALT
      MU1=MU1+1

      READ(NRIT2) (PL(J,L),(ERR(LL,J,L),LL=1,4),L=MU1,MU2)
      MU1=MU2
      MU2=MU2+NALTA
  331 CONTINUE
   11 CONTINUE
      IF(NOPT4.GT.1) CALL REDUCR(PL,NAZ,NALTA*NALT)
      CALL OTPNY (I,POA)
      GO TO (12,15),NOPT5
   12 CONTINUE
C       AZ  ACROSS
      L=0
      DO 14 J=1,NALT
      DO 13 K=1,NALTA
      L=L+1
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,400)ALT(J),ALTA(K),(PL(M,L),(ERR(LL,M,L),LL=1,4),M=1,NAZ)
   13 CONTINUE
   14 CONTINUE
      GO TO 50
   15 CONTINUE
C     AZ  DOWN
      L=0
      DO 16 J=1,NAZ
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,400)AZ(J),AINC(J),(PL(J,L),(ERR(LL,J,L),LL=1,4),L=1,1)
   16 CONTINUE
      GO TO 50
```

```
   20 CONTINUE
C     C3  DOWN
      DO 21 J=1,NAZ
      L=0
      DO 333 K=1,NC3
      L=L+1
      READ(NRIT2) POA(L),PL(J,L),(ERR(LL,J,L),LL=1,4)
  333 CONTINUE
   21 CONTINUE
      IF(NOPT4.GT.1) CALL REDUCR(PL,NAZ,NC3)
      CALL OTPNY (I,POA)
      GO TO (22,24),NOPT5
   22 CONTINUE
C     AZ ACROSS
      DO 23 J=1,NC3
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,500) POA(J),C3(J),(PL(K,J),(ERR(LL,K,J),LL=1,4),K=1,NAZ)
   23 CONTINUE
      GO TO 50
   24 CONTINUE
C     AZ  DOWN
      DO 25 J=1,NAZ
      WRITE (NRIT,400) AZ(J),AINC(J),(PL(J,K),(ERR(LL,J,K),LL=1,4),K=1,NC3)
   25 CONTINUE
   30 CONTINUE
   50 CONTINUE
      READ  (NRIT2) NERR
      IF(NERR .GT. 0) WRITE(NRIT,3000)
   60 CONTINUE
      RETURN
      END

$IBFTC SUB20
      SUBROUTINE PRPLY
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /ERRCOM/ ERR(4,10,10),NERR,PAERR
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      COMMON /ALACOM/ PL(50,10),POA(500)
      COMMON/WGTCOM/ GPFC,ANETP
  200 FORMAT(/14XF7.2,11X5(F10.2,4A1,1X))
  300 FORMAT(/9XF7.2,16X5(F10.2,4A1,1X))
  400 FORMAT(/9XF7.2,3XF7.2,6X5(F10.2,4A1,1X))
 3000 FORMAT(//,13X,37H* (XX) EXTRAPOLATION BEYOND DATA BANK)
      REWIND NRIT2
      DO 60 I=1,NOVEH
      READ(NRIT2) AME1,AME2,AME3,AME4
      READ(NRIT2) GPFC,ANETP
      IF(NOPT7.EQ.1.OR.I.EQ.1) CALL MISS
      CALL CAPT
```

```
      NYAW=NTYAW(I)
      GO TO (1,10,10,20,30),NOPT3
    1 CONTINUE
      CIRCULAR ORBIT
  100 FORMAT(7E15.8)
      DO 2 J=1,NYAW
      READ(NRIT2) (PL(J,L),(ERR(LL,J,L),LL=1,4),L=1,NALT)
    2 CONTINUE
      IF(NOPT4.GT.1) CALL REDUCR(PL,NYAW,NALT)
      CALL OTPNY(I,POA)
      GO TO (3,5),NOPT5
    3 CONTINUE
C     TIME YAW ACROSS
      DO 4 K=1,NALT
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,200) ALT(K),(PL(L,K),(ERR(LL,L,K),LL=1,4),L=1,NYAW)
    4 CONTINUE
      GO TO 50
    5 CONTINUE
      DO 6 K=1,NYAW
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,300) TYAW(K,I),(PL(K,L), (ERR(LL,K,L),LL=1,4),L=1,NALT)
    6 CONTINUE
      GO TO 50
   10 CONTINUE
C     ELLIPTICAL ORBITS
      DO 11 J=1,NYAW
      MU2=NALTA
      MU1=0
      DO 331 K=1,NALT
      MU1=MU1+1

      READ(NRIT2) (PL(J,L),(ERR(LL,J,L),LL=1,4),L=MU1,MU2)
      MU1=MU2
      MU2=MU2+NALTA
  331 CONTINUE
   11 CONTINUE
      IF(NOPT4.GT.1) CALL REDUCR(PL,NYAW,NALTA*NALT)
      CALL OTPNY(I,POA)
      GO TO (12,15),NOPT5
   12 CONTINUE
C     TIME  ACROSS
      L=0
      DO 14 J=1,NALT
      DO 13 K=1,NALTA
      L=L+1
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,400) ALT(J),ALTA(K),(PL(N,L),
     1(ERR(LL,N,L),LL=1,4),N=1,NYAW)
   13 CONTINUE
   14 CONTINUE
      GO TO 50
   15 CONTINUE
```

```
C     TIME YAW DOWN
      L=0
      DO 16 J=1,NYAW
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OTPNY(I,POA)
      WRITE(NRIT,400)TYAW(J,I),(PL(J,L),(ERR(LL,J,L),LL=1,4),L=1,NO)
   16 CONTINUE
      GO TO 50
   20 CONTINUE
C     C3
   30 CONTINUE
   50 CONTINUE
      READ  (NRIT2) NERR
      IF(NERR .GT. 0) WRITE(NRIT,3000)
   60 CONTINUE
      RETURN
      END

$IBFTC SUB21
      SUBROUTINE PRLLY
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON/BOSCOM/ TBOST(10),DTBOS(10),NTBOS(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      COMMON/ALACOM/ALA(20,5,5),ALO(20,5,5)
  200 FORMAT(/,11X,F7.2,17X,4(F7.2,1X,F7.2,3X))
      REWIND NRIT2
      NDUM=NALT
      IF(NOPT3.EQ.4)NDUM=NC3
      NDUM2=NAZ
      DO 50 I=1,NOVEH
      READ(NRIT2) AME1,AME2,AME3,AME4
      DT=DTBOS(I)
      NBOS=NTBOS(I)
      IF(NOPT2.EQ.2)NDUM2=NTYAW(I)
      IF(NOPT7.EQ.1.OR.I.EQ.1) CALL MISS
      CALL CAPT
      DO 2 K=1,NDUM2
      DO 7 K2=1,NDUM
      READ (NRIT2) (ALA(KK,K2,K),KK=1,NBOS)
      READ (NRIT2) (ALO(KK,K2,K),KK=1,NBOS)
    7 CONTINUE
    2 CONTINUE
      GO TO(3,5),NOPT5
    3 CONTINUE
C     AZ OR TYAW ACROSS
      DO 4 IJ=1,NDUM
      T=TBOST(I)-DT
      CALL OUTPLL(IJ,I)
```

```
      DO 4 IK=1,NBOS
      T=T+DT
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OUTPLL(IJ,I)
      WRITE(NRIT,200)T,(ALO(IK,IJ,IL),ALA(IK,IJ,IL),IL=1,NDUM2)
    4 CONTINUE
      GO TO 50
    5 CONTINUE
C     ALT OR C3 ACROSS
      DO 6 IJ=1,NDUM2
      T=TBOST(I)-DT
      CALL OUTPLL(IJ,I)
      DO 6 IK=1,NBOS
      T=T+DT
      CALL COUNT(1,1,NS)
      IF(NS.GT.0) CALL OUTPLL(IJ,I)
      WRITE(NRIT,200)T,(ALO(IK,IL,IJ),ALA(IK,IL,IJ),IL=1,NDUM)
    6 CONTINUE
   50 CONTINUE
      RETURN
      END

$IBFTC SUB22
      SUBROUTINE OTPNY(III,POA)
      DIMENSION NBLNK(5),NAPLD(23),NAPLE(23),NAPLF(23)
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      DIMENSION POA(10)
      DATA IBLNK/1H /
      DATA NBLNK/1,7,13,20,26/
      DATA NAPLD /1HC,1HU,1HT,1H ,1HO,1HF,1HF,1H ,1HW,1HE,1HI,1HG,1HH,1H
     1T,1H ,1H(,1HP,1HO,1HU,1HN,1HD,1HS,1H)/
      DATA NAPLE /1HG,1HR,1HO,1HS,1HS,1H ,1HP,1HA,1HY,1HL,1HO,1HA,1HD,1H
     1 ,1H ,1H(,1HP,1HO,1HU,1HN,1HD,1HS,1H)/
      DATA NAPLF /1HN,1HE,1HT,1H ,1HP,1HA,1HY,1HL,1HO,1HA,1HD,1H ,1H ,1H
     1(,1H ,1HP,1HO,1HU,1HN,1HD,1HS,1H ,1H)/
   10 FORMAT(//,3X,24HLAUNCH AZIMUTH (DEGREES),8X,5(F7.2,8X))
   11 FORMAT(3X,29HORBITAL INCLINATION (DEGREES),3X,5(F7.2,8X))
   20 FORMAT(/,10X,15HALTITUDE (N-MI),55A1)
   30 FORMAT(9X,17HPERIGEE    APOGEE)
   40 FORMAT(/,1X,13HPARKING ORBIT,9X,2HC3,55A1)
   41 FORMAT(1X,29HALTITUDE (N-MI)    (KM/SEC)**2)
   70 FORMAT(//,15X,15HALTITUDE (N-MI),5X,5(F7.2,8X))
   80 FORMAT(//,18X,15HALTITUDE (N-MI)/21X,7HPERIGEE,7X,5(F7.2,8X))
   81 FORMAT(21X,6HAPOGEE,8X,5(F7.2,8X))
   90 FORMAT(//,5X,29HPARKING ORBIT ALTITUDE (N-MI),1X,5(F7.2,8X))
   91 FORMAT(5X,15HC3  (KM/SEC)**2,15X,5(F7.2,8X))
  140 FORMAT(//,1X,29HTIME YAW STEERING BEGAN (SEC),5X,5(F7.2,8X))
```

```
776 FORMAT(/,9X,18HLAUNCH      ORBITAL/9X,22HAZIMUTH     INCLINATION/9X,
   1 16H(DEG)        (DEG),55A1)
780 FORMAT(/,1X,23HTIME YAW STEERING BEGAN ,55A1)
781 FORMAT(8X,9H(SECONDS))
    NYAW=NTYAW(III)
    GO TO (1,2),NOPT5
1   CONTINUE
C   ** AZ ACROSS ** ALT DOWN **
    GO TO (13,14),NOPT2
 13 WRITE(NRIT,10)(AZ(I),I=1,NAZ)
    WRITE(NRIT,11) (AINC(I),I=1,NAZ)
    NB=NBLNK(NAZ)
    GO TO 15
 14 WRITE(NRIT,140)(TYAW(II,III),II=1,NYAW)
    NB=NBLNK(NYAW)
 15 CONTINUE
    GO TO (3,3,3,4),NOPT3
  3 IF(NOPT4 .EQ.1) WRITE (NRIT,20) (IBLNK,I=1,NB),(NAPLD(I),I=1,23)
    IF(NOPT4 .EQ.2) WRITE (NRIT,20) (IBLNK,I=1,NB),(NAPLE(I),I=1,23)
    IF(NOPT4 .EQ.3) WRITE (NRIT,20) (IBLNK,I=1,NB),(NAPLF(I),I=1,23)
    GO TO (5,6,6),NOPT3
  5 RETURN
  6 WRITE(NRIT,30)
    RETURN
  4 IF(NOPT4 .EQ.1) WRITE (NRIT,40) (IBLNK,I=1,NB),(NAPLD(I),I=1,23)
    IF(NOPT4 .EQ.2) WRITE (NRIT,40) (IBLNK,I=1,NB),(NAPLE(I),I=1,23)
    IF(NOPT4 .EQ.3) WRITE (NRIT,40) (IBLNK,I=1,NB),(NAPLF(I),I=1,23)
    WRITE(NRIT,41)
    RETURN
  2 CONTINUE
C   ** AZ DOWN ** ALT ACROSS**
    NB=NBLNK(NALT)
    GO TO (7,8,8,9) ,NOPT3
  7 WRITE(NRIT,70) (ALT(I),I=1,NALT)
    GO TO 778
  8 WRITE(NRIT,80) (ALT(I),I=1,NALT)
    WRITE(NRIT,81) (ALTA(I),I=1,NALTA)
    GO TO 778
  9 WRITE(NRIT,90) (POA(I),I=1,NC3)
    WRITE(NRIT,91) (C3(I),I=1,NC3)
    NB=NBLNK(NC3)
778 CONTINUE
    GO TO(777,779),NOPT2
777 IF(NOPT4.EQ.1) WRITE(NRIT,776) (IBLNK,I=1,NB),(NAPLD(I),I=1,23)
    IF(NOPT4.EQ.2) WRITE(NRIT,776) (IBLNK,I=1,NB),(NAPLE(I),I=1,23)
    IF(NOPT4.EQ.3) WRITE(NRIT,776) (IBLNK,I=1,NB),(NAPLF(I),I=1,23)
    RETURN
779 CONTINUE
    IF(NOPT4.EQ.1) WRITE(NRIT,780) (IBLNK,I=1,NB),(NAPLD(I),I=1,23)
    IF(NOPT4.EQ.2) WRITE(NRIT,780) (IBLNK,I=1,NB),(NAPLE(I),I=1,23)
    IF(NOPT4.EQ.3) WRITE(NRIT,780) (IBLNK,I=1,NB),(NAPLF(I),I=1,23)
    WRITE(NRIT,781)
    RETURN
    END
```

```
$IBFTC SUB23
      SUBROUTINE OUTPLL(II,IV)
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      DIMENSION NAPLD(28),TITLE(2),IBLNK(5)
      DATA IBLNK/5,14,23,32,41/
      DATA ALTUD/100./
      DATA NBLNK/1H /
      DATA NAPLD/1HS,1HT,1HA,1HG,1HE,1H ,1HI,1HM,1HP,1HA,1HC,1HT,1H ,1HP
     1,1HO,1HI,1HN,1HT,1H ,1H(,1HD,1HE,1HG,1HR,1HE,1HE,1HS,1H)/
      DATA TITLE/6HLONG  ,6H  LAT /
   30 FORMAT(//,12X,15HALTITUDE (N-MI)/16X,F7.2)
   40 FORMAT(//,12X,15HALTITUDE (N-MI)/12X,15HPERIGEE  APOGEE/12X,F7.2,1
     1X,F7.2)
   50 FORMAT(//,2X,15HC3  (KM/SEC)**2,3X,29HPARKING ORBIT ALTITUDE (N-MI
     1)/6X,F7.2,18X,F7.2)
   80 FORMAT(/,3X,24HLAUNCH AZIMUTH (DEGREES),12X,4(F7.2,11X))
   81 FORMAT(3X,29HORBITAL INCLINATION (DEGREES),7X,4(F7.2,11X))
   90 FORMAT(/,3X,33HTIME YAW STEERING BEGAN (SECONDS),3X,4(F7.2,11X))
  100 FORMAT(/,5X,19HENGINE CUT OFF TIME,69A1)
  101 FORMAT(10X,9H(SECONDS),18X,4(A6,A6,6X))
  107 FORMAT(//,2X,55HLAUNCH AZIMUTH (DEGREES)  ORBITAL INCLINATION (DEG
     1REES)/10X,F7.2,20X,F7.2)
  120 FORMAT(//,3X,33HTIME YAW STEERING BEGAN (SECONDS)/16X,F7.2)
  130 FORMAT(/,19X,15HALTITUDE (N-MI),5X,4(F7.2,11X))
  140 FORMAT(/,19X,15HALTITUDE (N-MI)/23X,7HPERIGEE,9X,4(F7.2,11X))
  141 FORMAT(23X,6HAPOGEE,10X,4(F7.2,11X))
  150 FORMAT(/,7X,29HPARKING ORBIT ALTITUDE (N-MI),3X,4(F7.2,11X))
  151 FORMAT(21X,15HC3  (KM/SEC)**2,3X,4(F7.2,11X))
      NYAW=NTYAW(II)
      GO TO (1,2),NOPT5
    1 CONTINUE
C **  AZ ACROSS ALT FIXED
      GO TO (3,4,4,5,3),NOPT3
    3 CALL COUNT(20,5,NS)
      WRITE(NRIT,30)ALT(II)
      GO TO 7
    4 CALL COUNT(20,7,NS)
      WRITE(NRIT,40)ALT(II),ALTA(II)
      GO TO 7
    5 CALL COUNT(20,5,NS)
      WRITE(NRIT,50)C3(II),ALTUD
    7 CONTINUE
      GO TO (8,9),NOPT2
    8 WRITE(NRIT,80)(AZ(I),I=1,NAZ)
      WRITE(NRIT,81)(AINC(I),I=1,NAZ)
   82 CONTINUE
      NB=IBLNK(NAZ)
```

```
      NC=NAZ
      LC=LC+4
      GO TO 1000
    9 WRITE(NRIT,90) (TYAW(I,IV),I=1,NYAW)
      NB=IBLNK(NYAW)
      NC=NYAW
      LC=LC+2
      GO TO 1000
    2 CONTINUE
C *** ALT ACROSS AZ FIXED ***
      GO TO (11,12),NOPT2
   11 CONTINUE
      CALL COUNT(20,5,NS)
      WRITE(NRIT,107)AZ(II),AINC(II)
  109 CONTINUE
      GO TO (13,14,14,15,16),NOPT3
   13 CONTINUE
C **  CIRCULAR ORBIT
      WRITE(NRIT,130)(ALT(I),I=1,NALT)
      NC=NALT
      NB=IBLNK(NALT)
      LC=LC+2
      GO TO 1000
   14 WRITE(NRIT,140)(ALT(I),I=1,NALT)
      WRITE(NRIT,141)(ALTA(I),I=1,NALTA)
      NC=NALT
      NB=IBLNK(NALT)
      LC=LC+5
      GO TO 1000
   15 WRITE(NRIT,150)  ALT(1)
      WRITE(NRIT,151) (C3(I),I=1,NC3)
      NC=NC3
      NB=IBLNK(NC3)
      LC=LC+3
      GO TO 1000
   16 CONTINUE
C *** SYN ORBIT ***
      RETURN
   12 CONTINUE
C **  YAW STEERING **
      CALL COUNT(20,5,NS)
      WRITE(NRIT,120) TYAW(II,IV)
      GO TO 109
 1000 CONTINUE
      WRITE(NRIT,100)(NBLNK,I=1,NB),(NAPLD(I),I=1,28)
      WRITE(NRIT,101)(TITLE,I=1,NC)
      LC=LC+3
      RETURN
      END
```

```
$*
$IBFTC SUB24
      SUBROUTINE SYNCOT
      DIMENSION TITLE(2,4)
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON /ALACOM/ PL(500),A(500)
      COMMON/WGTCOM/ GPFC,ANETP
      COMMON /OPTCOM/ NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      DATA (TITLE(2,J),J=1,4)/4HDESC,4HENDI,4HNG N,4HODE /,
     1     (TITLE(1,J),J=1,4)/4HASCE,4HNDIN,4HG NO,4HDE  /
  100 FORMAT(//3X36HMISSION - SYNCHRONOUS ORBIT ALTITUDE)
  101 FORMAT(//3X38HMISSION - EQUITORIAL SYNCHRONOUS ORBIT)
  102 FORMAT(/3X10HVEHICLE - ,4A6)
  103 FORMAT(/3X21HLONGITUDE OF ARRIVAL ,F7.2,8H DEGREES)
  200 FORMAT(//3X19HDEPARTURE FROM THE ,4A4)
  300 FORMAT(/3X24HLAUNCH AZIMUTH (DEGREES),18X,5(F7.2,8X))
  301 FORMAT(/3X29HNUMBER OF PASSES THROUGH NODE,19X,I1,14X,I1,14X,I1,14
     1X,I1,14X,I1)
  302 FORMAT(/3X29HORBITAL INCLINATION (DEGREES),13X,5(F7.2,8X))
  400 FORMAT(/3X25HTIME FROM ORBIT INJECTION/3X30HTO FIRST DESCENDING NO
     1DE (SEC),10X,5(F9.2,6X))
  401 FORMAT(/3X32HTIME FROM FIRST DESCENDING NODAL/3X,39HPASSAGE TO DES
     1IRED DEPARTURE NODE (SEC),1X,5(F9.2,6X))
  500 FORMAT(/3X25HPAYLOAD TO ORBIT (POUNDS),14X,5(F10.2,4A1,1X))
  501 FORMAT(/3X32HCUT OFF WEIGHT TO ORBIT (POUNDS),7X,5(F10.2,4A1,1X))
  502 FORMAT(/3X31HGROSS PAYLOAD TO ORBIT (POUNDS),8X,5(F10.2,4A1,1X))
  503 FORMAT(/3X29HNET PAYLOAD TO ORBIT (POUNDS),10X,5(F10.2,4A1,1X))
  600 FORMAT(//)
 3000 FORMAT(//,13X,37H* (XX) EXTRAPOLATION BEYOND DATA BANK)
      REWIND NRIT2
      DO 7 I=1,NOVEH
      READ (NRIT2) AME1,AME2,AME3,AME4
      READ(NRIT2) GPFC,ANETP
      IF (NSYNC.EQ.1.OR.NSYNC.EQ.3) READ (NRIT2) JOLO,NOLO,NAZ
     1,(AZ(IJK),IJK=1,NAZ)
      IF (NSYNC.EQ.3) READ (NRIT2) (AINC(IJK),IJK=1,NAZ)
      READ(NRIT2) (PL(JK),(ERR(LL,JK),LL=1,4),JK=1,NAZ)
      IF(NOPT4.GT.1) CALL REDUCR(PL,NAZ,1)
      CALL PAGE
      IF(NSYNC.GT.2) WRITE(NRIT,100)
      IF(NSYNC.LT.3) WRITE(NRIT,101)
      WRITE(NRIT,102)AME1,AME2,AME3,AME4
      IF(NSYNC.EQ.2.OR.NSYNC.EQ.4) GO TO 5
      WRITE(NRIT,103)ALOND(I)
      IC=1
      ICNF=JOLO(1)
      DO 4 J=1,2
      IF(JOLO(J).EQ.0) GO TO 3
      WRITE(NRIT,200)(TITLE(J,K),K=1,4)
```

```
      WRITE(NRIT,300) (AZ(K),K=IC,ICNF)
      IF(NSYNC.EQ.3) WRITE(NRIT,302) (AINC(K),K=IC,ICNF)
      WRITE(NRIT,301) (NOLO (K),K=IC,ICNF)
      WRITE(NRIT,400) (TIME2(K),K=IC,ICNF)
      WRITE(NRIT,401) (TIME3(K),K=IC,ICNF)
    3 CONTINUE
      IC=JOLO(1)+1
      ICNF=NAZ
    4 CONTINUE
    5 CONTINUE
      WRITE(NRIT,600)
      WRITE(NRIT,300) (AZ(K),K=1,NAZ)
      IF(NSYNC.EQ.4) WRITE(NRIT,302) (AINC(K),K=1,NAZ)
    6 CONTINUE
C
C     WRITE(NRIT,500)((PL(K),A(IK,K),IK=1,4),K=1,NAZ)
C
      IF(NOPT4.EQ.1) WRITE(NRIT,501) (PL(K),(ERR(LL,K),LL=1,4),K=1,NAZ)
      IF(NOPT4.EQ.2) WRITE(NRIT,502) (PL(K),(ERR(LL,K),LL=1,4),K=1,NAZ)
      IF(NOPT4.EQ.3) WRITE(NRIT,503) (PL(K),(ERR(LL,K),LL=1,4),K=1,NAZ)
C
      READ  (NRIT2) NERR
      IF(NERR .GT. 0) WRITE(NRIT,3000)
    7 CONTINUE
      RETURN
      END


$IBFTC SUB25
      SUBROUTINE EXOUT
      COMMON /OPTCOM/ NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON/MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /RATCOM/ FTA(15),XETA(15),NPTA(3),NRTA(3)
     1,FTC(15),XETC(15),NPTC(3),NRTC(3)
     2,FISPA(15),XEISA(15),NPISA(3),NRISA(3)
     3,FISPC(15),XEISC(15),NPISC(3),NRISC(3)
     4,FSWA(15),XESWA(15),NPSWA(3),NRSWA(3)
     5,FSWC(15),XESWC(15),NPSWC(3),NRSWC(3)
      COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
     1,DELPR(3,3)
      COMMON /WHSCOM/ MSTG(3),MITM(3,3)
      COMMON /WGTCOM/ GPFC,ANETP
      COMMON /ERRCOM/ ERR(4,100),NERR,PAERR
      DIMENSION UNIT1(3),PAMA(3),PAMB(3),UNT2A(3),UNT2B(3),UNT3A(3),UNT3
     *B(3),UNT3C(3)
      DATA UNIT1/4HLB   ,4HSEC  ,4HLB   /
      DATA PAMA/4HTHRU,4HISP  ,4HSTAG/
      DATA PAMB/4HST   ,4H     ,4HE WT/
      DATA UNT2A/4HPERC,4HSEC  ,4HLB   /
      DATA UNT2B/4HENT  ,4H     ,4H     /
      DATA UNT3A/4HLB/P,4HLB/S,4HLB/L/
      DATA UNT3B/4HERCE,4HEC   ,4HB    /
```

```
      DATA UNT3C/4HNT   ,4H     ,4H     /
    3 FORMAT(/,9X,2A4,1X,F10.2,1X,A4,F10.2,1X,2A4,1XF7.2,1X,3A4,2X,F7.0
     *,3H LB)
  100 FORMAT(1H1, 13X60HMISSION-PAYLOAD DETERMINATION FOR LOW EARTH CIRC
     1ULAR ORBIT ,          /
     214X9HALTITUDE F7.2,1X6H(N-MI))
  200 FORMAT(1H1,  13X45HPAYLOAD DETERMINATION FOR HIGH ENERGY MISSION
     1/14X36HPARKING ORBIT ALTITUDE 100.00 (N-MI)
     2/14X3HC3 F7.2,1X11H(KM/SEC)**2)
  300 FORMAT(  14X10HVEHICLE     4A6)
  301 FORMAT(  14X15HLAUNCH AZIMUTH F7.2,2X9H(DEGREES)
     12X20HORBITAL INCLINATION F7.2,2X9H(DEGREES))
  302 FORMAT(//,32X,31H*** EXCHANGE RATIO ANALYSIS ***)
  303 FORMAT(//21X,7HNOMINAL,30X,7HPAYLOAD,12X,7HPAYLOAD)
  304 FORMAT(  2X,85HSTAGE  PARAMETER    VALUE           PERTURBATION
     *  EXCHANGE RATIO       PERTURBATION)
  305 FORMAT(/,4X,I1)
  306 FORMAT(/,75X,10H----------/69X,6HTOTAL ,F7.0,3H LB)
  307 FORMAT(60X,15HCUT OFF WEIGHT ,F7.0,3H LB,1X,4A1)
  308 FORMAT(61X,14HGROSS PAYLOAD ,F7.0,3H LB,1X,4A1)
  309 FORMAT(63X,12HNET PAYLOAD ,F7.0,3H LB,1X,4A1)
 3000 FORMAT(//,13X,37H* (XX) EXTRAPOLATION BEYOND DATA BANK)
      REWIND NRIT2
      DO 91 IJ=1,NOVEH
      READ(NRIT2) NSTGS
      READ(NRIT2)  AME1, AME2, AME3,AME4
      READ(NRIT2) GPFC,ANETP
      READ(NRIT2) ((VENOM(KI,JS),KI=1,3),JS=1,NSTGS)
      CALL WHSTG(IJ)
      GO TO (10,10,10,20,10),NOPT3
   10 CONTINUE
      WRITE (NRIT,100) ALT(1)
      GO TO 30
   20 CONTINUE
      WRITE (NRIT,200) C3(1)
   30 WRITE (NRIT,301) AZ(1),AINC(1)
      WRITE (NRIT,300)  AME1, AME2, AME3,AME4
      WRITE(NRIT,302)
      WRITE(NRIT,303)
      WRITE(NRIT,304)
      DO 92 J =1,NSTGS
      WRITE(NRIT,305) J
      DO 93  I=1,3
      IF((J.NE.MSTG(J)).OR.(I.NE.MITM(I,J))) GO TO 94
      READ(NRIT2) PLEX,DPLT
      WRITE(NRIT,3) PAMA(I),PAMB(I),VENOM(I,J),UNIT1(I)
     *,DELPR(I,J),UNT2A(I),UNT2B(I),PLEX
     *,UNT3A(I),UNT3B(I),UNT3C(I),DPLT
      GO TO 93
   94 CONTINUE
      WRITE(NRIT,3) PAMA(I),PAMB(I),VENOM(I,J),UNIT1(I)
   93 CONTINUE
   92 CONTINUE
      READ(NRIT2) DPAYS,PT,(ERR(II,1),II=1,4),NERR
      IF(NOPT4.GT.1)CALL REDUCR(PT,1,1)
```

```
      WRITE(NRIT,306) DPAYS
      IF(NOPT4.EQ.1)WRITE(NRIT,307) PT,(ERR(II,1),II=1,4)
      IF(NOPT4.EQ.2) WRITE(NRIT,308) PT,(ERR(II,1),II=1,4)
      IF(NOPT4.EQ.3) WRITE(NRIT,309) PT,(ERR(II,1),II=1,4)
      IF(NERR .GT. 0) WRITE(NRIT,3000)
   91 CONTINUE
      RETURN
      END

$IBFTC SUB26
      SUBROUTINE INFO
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      DIMENSION XNUM(6),A(2),B(2)
      DATA XNUM/1H1,1H2,1H3,1H4,1H5,1H /,A/1H,,1H /,B/2H**,2H  /
    1 FORMAT(1H1,42X,48HTHE DATA BANK CONTAINS THE FOLLOWING INFORMATION
     1)
    2 FORMAT(//,26X,7HVEHICLE,28X,9HMISSION *,24X,26HEXCHANGE RATIO FUNC
     1TION OF/100X,14HALTITUDE     C3)
    3 FORMAT(/26X,9HSATURN IB,26X,9A1,33X,A2,7X,A2)
    4 FORMAT(26X,12HSATURN V 516,23X,9A1,33X,A2,7X,A2)
    5 FORMAT(26X,16HSATURN V 2 STAGE,19X,9A1,33X,A2,7X,A2)
    6 FORMAT(26X,11HTITAN III C,24X,9A1,33X,A2,7X,A2)
    7 FORMAT(26X,11HTITAN III M,24X,9A1,33X,A2,7X,A2)
    8 FORMAT(//,53X16H* MISSION LEGEND//53X,18H1  LOW EARTH ORBIT/53X,14
     1H2  POLAR ORBIT/53X,14H3  HIGH ENERGY/53X,20H4  SYNCHRONOUS ORBIT)
      WRITE(NRIT,1)
      WRITE(NRIT,2)
      WRITE(NRIT,3) XNUM(1),A(1),XNUM(2),A(1),XNUM(3),A(2),XNUM(6),A(2),
     1XNUM(6),B(1),B(2)
      WRITE(NRIT,4) XNUM(1),A(1),XNUM(3),A(2),XNUM(6),A(2),XNUM(6),A(2),
     1XNUM(6),B(2),B(1)
      WRITE(NRIT,5) XNUM(1),A(1),XNUM(2),A(1),XNUM(3),A(2),XNUM(6),A(2),
     1XNUM(6),B(1),B(2)
      WRITE(NRIT,6) XNUM(1),A(2),XNUM(6),A(2),XNUM(6),A(2),XNUM(6),A(2),
     1XNUM(6),B(2),B(2)
      WRITE(NRIT,7) XNUM(1),A(2),XNUM(6),A(2),XNUM(6),A(2),XNUM(6),A(2),
     1XNUM(6),B(2),B(2)
      WRITE(NRIT,8)
      RETURN
      END

$IBFTC SUB27
      SUBROUTINE MISS
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /YAWCOM/ TYAW(5,10),NTYAW(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
      COMMON /LINCOM/ LC,LMAX,NPAGE
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
   40 FORMAT(/,27X,60HMISSION - PAYLOAD DETERMINATION FOR LOW EARTH CIRC
     1ULAR ORBIT )
```

```
   50 FORMAT(/,27X,62HMISSION - PAYLOAD DETERMINATION FOR LOW EARTH ELLI
     1PTICAL ORBIT)
   60 FORMAT(/,27X,62HMISSION - PAYLOAD DETERMINATION FOR LOW EARTH ELLI
     1PTICAL ORBIT/37X,33HAND CIRCULARIZING ORBIT AT APOGEE)
   70 FORMAT(/,27X,56HMISSION - PAYLOAD DETERMINATION FOR HIGH ENERGY MI
     1SSIONS)
   80 FORMAT(/,27X,36HMISSION - SYNCHRONOUS ORBIT ALTITUDE)
   81 FORMAT(/,27X,45HMISSION - EQUITIORIAL EARTH SYNCHRONOUS ORBIT)
   90 FORMAT(/,27X,63HMISSION - STAGE IMPACT POINT TRACE FOR LOW EARTH C
     1IRCULAR ORBIT)
  100 FORMAT(/,27X,65HMISSION - STAGE IMPACT POINT TRACE FOR LOW EARTH E
     1LLIPTICAL ORBIT)
  110 FORMAT(/,27X,59HMISSION - STAGE IMPACT POINT TRACE FOR LOW EARTH E
     1LLIPTICAL/37X,33HORBIT AND CIRCULARIZING AT APOGEE)
  120 FORMAT(/,27X,59HMISSION - STAGE IMPACT POINT TRACE FOR HIGH ENERGY
     1 MISSIONS)
      CALL PAGE
      LC=2
      GO TO(1,2,3),NOPT1
    1 CONTINUE
C*****PAYLOAD DETERMINATION*****
      GO TO (4,5,6,7,8),NOPT3
    4 WRITE(NRIT,40)
      RETURN
    5 WRITE(NRIT,50)
      RETURN
    6 WRITE(NRIT,60)
      RETURN
    7 WRITE(NRIT,70)
      RETURN
    8 CONTINUE
      RETURN
    2 CONTINUE
C*****STAGE IMPACT POINT TRACE*****
      GO TO (9,10,11,12,13),NOPT3
    9 WRITE(NRIT,90)
      RETURN
   10 WRITE(NRIT,100)
      RETURN
   11 WRITE(NRIT,110)
      RETURN
   12 WRITE(NRIT,120)
      RETURN
   13 CONTINUE
      IF(NSYNC.GT.2) WRITE(NRIT,80)
      IF(NSYNC.LT.3) WRITE(NRIT,81)
      RETURN
    3 CONTINUE
      RETURN
      END
```

```
$IBFTC SUB28
      SUBROUTINE CAPT
      COMMON/OPTCOM/NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME1,AME2,AME3,AME4
    1 FORMAT(/27X10HVEHICLE -       4A6)
  100 FORMAT(/,27X15HLAUNCH AZIMUTH F7.2,7H  (DEG)/27X33HORBITAL INCLINA
     1TION  90.00  (DEG))
      CALL COUNT(10,2,NS)
      WRITE (NRIT,1) AME1,AME2,AME3,AME4
      IF(NOPT2.EQ.1) RETURN
      CALL COUNT(8,2,NS)
      WRITE(NRIT,100) AZ(1)
      RETURN
      END


$IBFTC SUB29
      SUBROUTINE MATAZ (AZ)
      COMMON /ROTCOM/ T90(3,3),T90I(3,3),TAZ(3,3),ABC(3,3)
      CALL ROT(((AZ-90.)/57.2957795),TAZ)
      CALL MAT (T90I,TAZ,ABC,3,3,3)
      CALL MAT (ABC,T90,TAZ,3,3,3)
      RETURN
      END


$IBFTC SUB30
      SUBROUTINE ROT(ANG,A)
      DIMENSION A(3,3)
C     ANG IN RADIANS
      C=COS(ANG)
      S=SIN(ANG)
      DO 1 I=1,3
      DO 1 J=1,3
      A(J,I)=0.
    1 CONTINUE
C     ROTATION ABOUT THE X-AXIS
      A(1,1)=1.
      A(2,2)=C
      A(2,3)=S
      A(3,2)=-S
      A(3,3)=C
      RETURN
      END


$IBFTC SUB31
      SUBROUTINE MAT (P,Q,R,J,K,L)
C     R(J,L)=P(J,K)*Q(K,L)
      DIMENSION P(3,3),Q(3,3),R(3,3),QQ(3,3)
      DO 1 LL=1,L
```

```
      DO 1 KK=1,K
      QQ(KK,LL)=Q(KK,LL)
    1 CONTINUE
      DO 2 LL=1,L
      DO 2 JJ=1,J
      R(JJ,LL)=0.0
    2 CONTINUE
      DO 3 LL=1,L
      DO 3 JJ=1,J
      DO 3 KK=1,K
      R(JJ,LL)=R(JJ,LL)+P(JJ,KK)*QQ(KK,LL)
    3 CONTINUE
      RETURN
      END


$IBFTC SUB32
      SUBROUTINE INTLA(NNN,NDIM,F,X,NX,NOR,C,XX,ANS)
      DIMENSION F(1),X(1),XX(1),C(1),NX(4),NS(4),NF(4),NOR(4)
      NS(2)=1
      NS(3)=1
      NS(4)=1
      L=1
      DO 3 I=1,NDIM
      K=NX(I)
      IF (I.GT.NNN) GO TO 1
C     CURVE FIT
      CALL CFCOF (L,K,XX(I),C(L),NS(I),NF(I))
      GO TO 2
    1 CONTINUE
C     LAGRANGE COEFS
      CALL LACOF (X(L),K,XX(I),NOR(I),C(L),NS(I),NF(I))
    2 CONTINUE
      L=L+K
    3 CONTINUE
      IXS=NS(1)
      IXF=NF(1)
      JYS=NS(2)
      JYF=NF(2)
      KZS=NS(3)
      KZF=NF(3)
      LVS=NS(4)
      LVF=NF(4)
      IX=NX(1)
      JY=NX(2)
      KZ=NX(3)
      IXY=IX*JY
      IXJ=IX*(JYS-1)
      IXYZ=IXY*KZ
      LFX=IX-IXF
      LFY=(JY-JYF)*IX
      NSS=IX+JY+KZS-1
      III=IXY*(KZS-1)
      NL=IX+JY+KZ+LVS-1
      MS=IX+JYS-1
```

```
      ANSV=0.
      DO 40 LLV=LVS,LVF
      N=NSS
      L=(LLV-1)*IXYZ+III
      ANSK=0.
      DO 30 K=KZS,KZF
      L=L+IXJ
      ANSJ=0.
      M=MS
      DO 20 J=JYS,JYF
      L=L+IXS-1
      ANS=0.
      DO 10 I=IXS,IXF
      L=L+1
      ANS=ANS+C(I)*F(L)
   10 CONTINUE
      IF (NDIM.EQ.1) RETURN
      M=M+1
      ANSJ=ANSJ+C(M)*ANS
      L=L+LFX
   20 CONTINUE
      ANS=ANSJ
      IF (NDIM.EQ.2) RETURN
      N=N+1
      ANSK=ANSK+C(N)*ANS
      L=L+LFY
   30 CONTINUE
      ANS=ANSK
      IF (NDIM.EQ.3) RETURN
      NL=NL+1
      ANSV=ANSV+C(NL)*ANS
   40 CONTINUE
      ANS=ANSV
      RETURN
      END

$IBFTC SUB33
      SUBROUTINE CFCOF (L,K,XX,C,NS,NF)
      DIMENSION C(1)
      C(1)=1.
      DO 1 I=2,K
      C(I)=C(I-1)*XX
    1 CONTINUE
      NS=L
      NF=L+K-1
      RETURN
      END

$IBFTC SUB34
      SUBROUTINE LACOF (X,NX,XX,NOR,C,I,N)
      DIMENSION X(1),C(1)
      NO=NOR
      IF (NO.GT.NX) NO=NX-1
      I=NX+1
      DO 10 J=1,NX
```

```
      I=I-1
      IF (XX.GE.X(I)) GO TO 20
   10 CONTINUE
   20 CONTINUE
      N=I+NO
      IF (N.LE.NX) GO TO 30
      N=NX
      I=NX-NO
   30 CONTINUE
      DO 50 K=I,N
      S=1.
      SS=1.
      AX=X(K)
      DO 40 L=I,N
      IF (K.EQ.L) GO TO 40
      S=S*(XX-X(L))
      SS=SS*(AX-X(L))
   40 CONTINUE
      C(K)=S/SS
   50 CONTINUE
      RETURN
      END

$IBFTC SUB35
      SUBROUTINE C3ALT (NS,AP,AA,C3)
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
C     NS=1  C3=F(ALT)
C     NS=2 ALT=G(C3) CIRCULAR ORBITS ONLY
      GO TO (1,2),NS
    1 CONTINUE
      C3=-2.*U/(2.*RE+GNMK*(AP+AA))
      RETURN
    2 CONTINUE
      AP=-U/C3
      AA=AP
      RETURN
      END


$IBFTC SUB36
      SUBROUTINE INCAZ (NS,NN)
      DIMENSION X(5),AZINC(5)
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      DATA X/45.,70.,90.,120.,145./
      DATA AZINC/49.468,33.396,28.39,39.852,58.338/
C     NS=1 NORTH NS=2 SOUTH
C     NN=0 AZ INC  NN=1 INC TO AZ
      N=1
      IF (NS.NE.1) N=3
      DO 2 I=1,NAZ
      IF (NN.EQ.0) GO TO 1
      CALL INTLA (0,1,X(N),AZINC(N),3,2,C,AINC(I),AZ(I))
      GO TO 2
```

```
    1 CONTINUE
      CALL INTLA (0,1,AZINC(N),X(N),3,2,C,AZ(I),AINC(I))
    2 CONTINUE
      RETURN
      END

$IBFTC SUB37
      SUBROUTINE WHSTG(IVEH)
      COMMON /WHSCOM/ MSTG(3),MITM(3,3)
      COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
     1,DELPR(3,3)
      DO 10 I=1,3
      MSTG(I)=0
      DO 10 J=1,3
      MITM(J,I)=0
   10 CONTINUE
      JSTG=NSTAGS(IVEH)
      DO 30 I=1,JSTG
      ISTG=NSTAG(I,IVEH)
      MSTG(ISTG)=ISTG
      KITMS=ITEMS(I,IVEH)
      DO 30 J=1,KITMS
      KITM=ITEM(J,I)
      MITM(KITM,ISTG)=KITM
   30 CONTINUE
      RETURN
      END

$IBFTC SUB38
      SUBROUTINE ANUMB (I,NI,N)
      DIMENSION NI(3)
      N=1
      IF (I.EQ.1) RETURN
      K=I-1
      DO 1 J=1,K
      N=N+NI(J)
    1 CONTINUE
      RETURN
      END

$IBFTC SUB39
      SUBROUTINE REDUCR(PL,N,M)
      COMMON /WGTCOM/ GPFC,ANETP
      COMMON /OPTCOM/ NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,NOPT6,NOPT7
C     GPFC - GROSS WT FROM CUT OFF WT
C     ANETP - NET PAYLOAD FROM GROSS WT
      DIMENSION PL(1,1)
      A=GPFC
      IF(NOPT4.GT.2) A=A+ANETP
      DO 1 I=1,N
      DO 1 J=1,M
      PL(I,J)=PL(I,J)-A
    1 CONTINUE
      RETURN
      END
```

```
$IBFTC SUB40
      SUBROUTINE COUNT(NP,NPA,NS)
      COMMON/LINCOM/LC,LMAX,NPAGE
      NS=0
      IF((NP+LC).LT.LMAX) GO TO 2
      LC=0
      NS=1
      CALL PAGE
    2 CONTINUE
      LC=LC+NPA
      RETURN
      END

$IBFTC SUB41
      SUBROUTINE PAGE
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON/LINCOM/LC,LMAX,NPAGE
      NPAGE=NPAGE+1
      WRITE(NRIT,100) NPAGE
  100 FORMAT(1H1,/100X,4HPAGE,I4)
      RETURN
      END

$IBFTC SUB42
      SUBROUTINE AZDET (I)
      DIMENSION FLO(9),XDL(45),XAL(45)
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      DATA NXLO/9/,NORLO/6/
      DATA FLO /125.,120.,110.,100.,90.,80.,70.,60.,55./
C     XDL - DEPARTURE FROM DESCENDING NODE
      DATA XDL / 312.5,317.,327.5,342.,362.5,382.,397.,409.5,415.
     1,           290.,295.,305.5,320.,340.,359.5,375.,387.,393.
     2,           268.5,273.,283.,297.,317.,336.,352.5,365.,370.5
     3,           246.,250.,261.,275.,295.,314.,330.,342.5,348.5
     4,           224.,228.,238.,252.,272.5,292.,307.5,320.,325./
C     XAL - DEPARTURE FROM ASCENDING NODE
      DATA XAL / 122.,126.5,137.,152.,172.,191.,206.,218.5,224.
     1,           100.,104.,114.5,129.5,149.5,167.,183.,196.,201.5
     2,           77.,81.5,92.,107.,127.,146.,161.,174.,179.5
     3,           56.,60.,70.,84.5,104.,123.,138.,150.,156.5
     4,           32.,36.5,47.,62.,81.,100.,115.,128.,134./
C     ALOND LONG OF ARRIVAL
C     ALOD  LONG OF DEPARTURE
      ALOD=ALOND(I)-100.
      IF (NSYNC.GE.3) ALOD=ALOND(I)-79.03
      IF (ALOD.LT.0.) ALOD=ALOD+360.
      NAZ=0
C     ASCENDING NODE DEPARTURE
      CALL AMACH (1.,ALOD,FLO,XAL,NXLO,NORLO,1)
C     DESCENDING NODE DEPARTURE
```

```
      ALONF=ALOD
      IF (ALOD.LE.60.) ALONF=ALONF+360.
      CALL AMACH (0.,ALONF,FLO,XDL,NXLO,NORLO,2)
      RETURN
      END

$IBFTC SUB43
      SUBROUTINE AMACH (A,ALD,F,X,NX,NOR,J)
      DIMENSION F(1),X(1),XAZ(7),CAF(7)
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      COMMON /MISCOM/ AZ(10),NAZ,ALT(10),NALT,ALTA(10),NALTA,C3(10),NC3
     1,AINC(10),NOVEH,NOVE(10)
     1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
      COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
      EQUIVALENCE (K,NAZ)
      DATA XAZ /60.,70.,80.,90.,100.,110.,120./
      DATA CAF /131.5,121.,107.2,90.,73.5,59.,49./
C     A=1 DEPARTURE FROM ASCENDING NODE
C     A=0 DEPARTURE FROM DESCENDING NODE
      JOLO(J)=0
      N=-8
      DO 1 I=1,5
      N=N+9
      IF (ALD.LT.X(N).OR.ALD.GT.X(N+8)) GO TO 1
      JOLO(J)=JOLO(J)+1
      K=K+1
      NOLO(K)=I
      CALL INTLA (0,1,F,X(N),NX,NOR,C,ALD,AZ(K))
      TIME3(K)=5288.*(.5*A+FLOAT(I-1))
      CALL INTLA (0,1,CAF,XAZ,7,6,C,AZ(K),CA)
      TIME2(K)=14.69*(CA-22.)
    1 CONTINUE
      RETURN
      END

$IBFTC SUB44
      SUBROUTINE ERROR(ND,F,X,NX,NORX,XX,A,N1,N2)
      DIMENSION A(4),NX(1),F(1),X(1),NORX(1),XX(1),AA(2),BB(2),B(10)
     1,C(2),A1(2),B1(2)
      COMMON /ERRCOM/ ERR(400),NERR,PAERR
      DATA B/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/,C/1H(,1H)/,D/1H /
     1,AA/.059,.055/,BB/.061,.057/
      DO 1 I=1,4
      A(I)=D
    1 CONTINUE
      DO 2 I=1,2
      A1(I)=0.0
      B1(I)=0.0
    2 CONTINUE
      L=1
      ERROR=0.0
      PAERR=2.00
      DO 3 I=N1,N2
      K=NX(I)
      K1=L+K-1
```

```
      IF(XX(I).LT.X(L)) A1(I)=BB(I)
      IF(XX(I).GT.X(K1)) B1(I)=AA(I)
      ERROR=ERROR + (X(L)-XX(I))* A1(I) + (XX(I)-X(K1))* B1(I)
      L=L+K
    3 CONTINUE
      IF(ERROR .LE.PAERR) RETURN
      IF(ERROR.GT.99.) ERROR= 99.00
      NERR=NERR + 1
      A(1)=C(1)
      A(4)=C(2)
      M=ERROR
      N=0.1*ERROR
      LL=M-N*10
      A(2)=B(N+1)
      A(3)=B(LL+1)
      RETURN
      END

$IBFTC SUB45
      SUBROUTINE CURVE
      DIMENSION TIME(100),TRACE(100),COEF(7)
     Z,ACO(2),BCO(2),CCO(2),DAT(14)
      COMMON/CSTCOM/U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
C
      NAMELIST /ONE/ TIME,TRACE,NSET,NP
C
      DATA ACO /4HLATI,4HLONG/
      DATA BCO /4HTUDE,4HITUD/
      DATA CCO /4H    ,4HE   /
      DATA DAT/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD,1HE/
C
   10 FORMAT(1H1,5X50HTHE COEFFICIENTS GENERATED IN THIS SUBROUTINE
     1  /55H WILL BE PUNCHED ON CARDS IN DATA STATEMENTS TO BE USED
     2  /55H BY THE PROGRAM. ALL THE USER NEED DO IS PUNCH THE
     3  /55H CORRECT NUMBER FOLLOWING FY ON THE FIRST CARD AND
     4  /55H THE LIMITS ON I. CONSULT THE SECTION DESCRIBING
     5  55H DATA BLOCKS IN THE DECK WRITE UP.                      )
   11 FORMAT(6X10HDATA FY  /   )
   12 FORMAT(1X4A4)
   13 FORMAT(1X6HTIME  9(1XF9.2))
   14 FORMAT(1X6HTRACE 9(1XF9.2))
   15 FORMAT(1X,3H A0E15.8,3X3HA1 E15.8,3X3HA2 E15.8,3X3HA3 E15.8/
     13H A4E15.8,3X3HA5 E15.8,3X3HA6 E15.8,3X3HA7 E15.8/3H A8E15.8,
     23HA9 E15.8)
   16 FORMAT(5X1A1,1X4(E15.8,1H,))
   17 FORMAT(5X1A1,1XE15.8,1H,,E15.8,1H,,E15.8,1H/)
C
      WRITE (NRIT,10)
C
      NPNCH=7
      READ (NRED,ONE)
      DO 30 I=1,2
C     I=1 LAT   I=2 LONG
      WRITE (NPNCH,11)
      WRITE (NRIT,12) ACO(I),BCO(I),CCO(I)
```

```
      L=0
      DO 20 J=1,NSET
      READ (NRED,ONE)
      L=L+1
      WRITE (NRIT,13) (TIME(K),K=1,NP)
      WRITE (NRIT,14) (TRACE(K),K=1,NP)
      CALL CFIT (TIME,TRACE,NP,6,COEF)
      WRITE (NRIT,15) COEF
      WRITE (NPNCH,16) DAT(L),(COEF(M),M=1,4)
      L=L+1
      IF (J.LT.NSET) WRITE (NPNCH,16) DAT(L),(COEF(M),M=5,7)
      IF (J.GE.NSET) WRITE (NPNCH,17) DAT(L),(COEF(M),M=5,7)
      IF (L.GE.14) L=0
   20 CONTINUE
   30 CONTINUE
      RETURN
      END

$IBFTC SUB46
      SUBROUTINE CFIT( X , Y ,NP,ND1,COEF)
C
C    X    ARRAY OF INDEPENT VARUABLES
C    Y    ARRAY OF DEPENDENT VARIABLES
C    NP    NO. OF POINTS IN AID AND ADV
C    NDI    DEGREE OF CURVE FIT
C    COE     ARRAY OF COEFFICIENTS
C    SE    STANDARD ERROR OF ESTIMATE
C    P    INDEX OF MULTIPLE CORRELATION
C   P2  INDEX OF DETERMINATION
C   SY    STANDARD DEVIATION
C    NSL    SOLUTION INDICATOR    MUST BE ZERO
C
      DIMENSION   X(100),  Y(100),         SUM(19),SUMY(10),COE(10)
      NSL=0
      CALL LSQ( X , Y ,  SUM,SUMY,COE,ND1,NSL)
      IF (NSL)7,10,7
    7 RETURN
   10 NX=ND1+1
      SUM1=0.
      SUM2=0.
      SUM3=0.
      YBAR=0.
      XBAR=0.
      DO 13 J=1,NP
      YC=COE(1)
      DO 12 K=2,NX
      YC=YC+COE(K)*X(J)**(K-1)
   12 CONTINUE
      DIF=Y(J)-YC
      SUM1=SUM1+DIF**2
      SUM2=SUM2+ABS(DIF)
      YBAR=YBAR+Y(J)
      XBAR=XBAR+X(J)
   13 CONTINUE
      FN=NP
      FR=NX
```

```
      SE=SQRT(SUM1/(FN-FR))
      FMD=SUM2/FN
      YBAR=YBAR/FN
      XBAR=XBAR/FN
      DO 14 J=1,NP
      DIF=Y(J)-YBAR
      SUM3=SUM3+DIF**2
   14 CONTINUE
      P2=1.-(SUM1*(FN-1.))/(SUM3*(FN-FR))
      P=SQRT(P2)
      SY=SQRT(SUM3/(FN-1.))
      RETURN
      END

$IBFTC SUB47
      SUBROUTINE LSQ(X,Y,  SUM,SUMY,COE,ND1,NSL)
      DIMENSION X(100),Y(100),A(10,11),SUM(19),SUMY(10),
     1COE(10)
      DO 10 I=1,19
   10 SUM(I)=0.
      DO 11 I=1,10
   11 SUMY(I)=0.
      N=2*ND1+1
      DO 30 I=1,N
      DO 30 L=1,NP
      IF(X(L))17,15,17
   15 IF(I-1)17,16,17
   16 ER=1.
      GO TO 18
   17 ER=X(L)**(I-1)
   18 CONTINUE
      SUM(I)=SUM(I)+ER
      IF(I-ND1-1)20,20,30
   20 SUMY(I)=ER*Y(L)+SUMY(I)
   30 CONTINUE
      NX=ND1+1
      DO 50 I=1,NX
      M=I
      DO 40 J=1,NX
      A(I,J)=SUM(M)
   40 M=M+1
      A(I,J)=SUMY(I)
   50 CONTINUE
      CALL SIMEQ(A,NX,COE,ND1,NSL)
      RETURN
      END

$IBFTC SUB48
      SUBROUTINE SIMEQ(A,N,C,ND1,NSL)
      COMMON /CSTCOM/ U,G,RE,CNMK,PIE,RAD,C(50),NRED,NRIT,NRIT2
      DIMENSION A(10,11),B(10,11)
      TOL=5.0E-08
      KA=N+1
      K2=2
      K3=1
```

```
      DO 10 I=1,N
      DO 10 J=1,KA
C     MOVE MATRIX TO WORKING AREA
   10 B(I,J)=A(I,J)
   20 IF(ABS(B(K3,K3))-TOL)30,100,100
   30 IF(K2-N)40,40,300
   40 DO 50 I=1,KA
      D=B(K3,I)
      B(K3,I)=B(K2,I)
      B(K2,I)=D
   50 CONTINUE
      K2=K2+1
      GO TO 20
  100 K1=K3
      K3=K3+1
      K2=K3+1
      IF(K3-N)120,120,200
  120 DO 130 I=K3,KA
      B(K1,I)=B(K1,I)/B(K1,K1)
  130 CONTINUE
      DO 140 I=K3,N
      DO 140 J=K3,KA
      B(I,J)=B(I,J)-B(I,K1)*B(K1,J)
  140 CONTINUE
      GO TO 20
  200 C(N)=B(N,N+1)/B(N,N)
      K3=N-1
  210 K1=K3+1
      D=0.
      DO 220 I=K1,N
  220 D=D+B(K3,I)*C(I)
      C(K3)=B(K3,KA)-D
      K3=K3-1
      IF(K3)350,350,210
  300 TOL=TOL/10.
      IF(TOL-1.0E-30)320,310,310
  310 K2=K3+1
      GO TO 20
  320 WRITE (NRIT,321)ND1
      NSL=1
  321 FORMAT (//35H NO SOLUTION FROM SIMEQ FOR DEGREE I2//)
  350 RETURN
      END


$ORIGIN        ORG1
$IBFTC SUB49
       BLOCK DATA
C          THIS BLOCK COUNTAINS DATA FOR VHICLE NO. 4, SA 212 PMR J2.
C          IT WAS FORMED IN JUNE 1969 FROM TRAJECTORIES GENERATED ON
C          THE ROBOT II COMPUTER PROGRAM.
C
C          THESE LABEL COMMON MUST BE PRESENT IN THE BLOCK,
C          FCNCOM, XPCOM, XYWCOM, C3COM, ECCOM AND VEHCOM
```

```
C         RATCOM,SYNCOM,WDGCOM
C
       COMMON /FCNCOM/ F1(25),F2(35),F3(35),F4(70),F5(70)
      1,FY1(25),FY2(105),FY3(105),FY4(105),FY5(105)
      2,FY11(25),FY12(105),FY13(105),FY14(105),FY15(105)
       COMMON /XPCOM/ X(10),NX(2),NORX(2),XLLP(16),NXLLP(3),NORLLP(3)
      1,XLLC3(15),NXLLC3(3),NORLC3(3)
       COMMON /XYWCOM/ XYP1(8),NXYP1(2),NORYP1(2)
      1,XYP2(8),NXYP2(2),NORYP2(2)
      2,XYLLP1(15),NXYLP1(3),NRYLP1(3)
      3,XYLLP2(15),NXYLP2(3),NRYLP2(3)
       COMMON /C3COM/ SLOPC(15),XC3(15),NXC3(2),NORC3(2)
      1,SLOPAL(5),NSOPAL,NORPAL,C3COFF
       COMMON /ECCOM/ SLOPE(5),NXE,NORE,SOPEY1(5),NXEY1,NOEY1
      1,SOPEY2(5),NXEY2,NOEY2
       COMMON /VEHCOM/ NSTGS,VENOM(3,3),AME(4)
       COMMON /RATCOM/ FTA(15),XETA(15),NPTA(3),NRTA(3)
      1,FTC(15),XETC(15),NPTC(3),NRTC(3)
      2,FISPA(15),XEISA(15),NPISA(3),NRISA(3)
      3,FISPC(15),XEISC(15),NPISC(3),NRISC(3)
      4,FSWA(15),XESWA(15),NPSWA(3),NRSWA(3)
      5,FSWC(15),XESWC(15),NPSWC(3),NRSWC(3)
       COMMON /ITMCOM/ NSTAGS(10),NSTAG(3,10),ITEMS(3,10),ITEM(3,3)
      1,DELPR(3,3)
       COMMON /WGTCOM/ GPFC,ANETP
       COMMON/SYNCOM/NSYNC,JOLO(2),NOLO(10),ALOND(10),TIME2(10),TIME3(10)
      1, DVSYA,DVSYN,XSAZ(10),NXSY,NRSY,XLSAZ(12),NXLSY(2),NLRSY(2)
       DATA AME/6HSA 212,6H PMR   ,6H  J2    ,6H           /
C         F1 CONTAINS THE CUTOFF WTS. TO THE ALTS. AND AZS.
C         CONTAINED IN X
C   NX(1)=NO. OF POINTS IN AZ , NX(2)=NO. OF PTS. IN ALT
C   NORX(1)=ORDER OF INTERPOLATION IN AZ, NORX(2)=ORDER IN ALT
       DATA NX/5,5/,NORX/3,4/
      1,X/45.,70.,90.,120.,145.,100.,200.,300.,400.,500./
      2,F1/66876.,68732.,69241.,68220.,65965.
      3,59299.,60843.,61265.,60419.,58542.
      4,49729.,50905.,51227.,50583.,49149.
      5,39299.,40122.,40347.,39898.,38893.
      6,29116.,29643.,29787.,29500.,28857./
C SLOPE CONTAINS SLOPES OF LINES FOR DECREASE IN COW FOR VARIOUS
C   PERIGEE ALTS. AND A FIXED APOGEE ALT.
       DATA NXE/5/,NORE/3/,SLOPE/-8.6025,-7.45,-5.82,-4.13,0./
       DATA NXLLP/7,5,1/,NORLLP/7,3,1/,XLLP/7*0.,100.,200.,
      1300.,400.,500.,4*0./
C         F2 CONTAINS COEFFICIENTS OF POLYNOMIALS DEFINING LAT. OF
C         IMPACT AS A FUNCTION OF TIME FOR VARIOUS ALTITUDES.
       DATA F2/27.157635,.048952316,-.68317544E-3,.43395832E-5,        B   100
      1-.14111942E-7,.225979000E-10,-.14257947E-13,                    B   100
      226.603738,.70658806E-1,-.97876247E-3,-.62237778E-5,-.20275392E-7, B   200
      3.32594400E-10,-.20750461E-13 ,                                  B   200
      4.26766697E+2,.65759853E-1,-.94065166E-3,.61807018E-5,           B   300
      5-.20756495E-7,.34351576E-10,-.22590349E-13,                     B   300
      626.79605,.62986912E-1,-.87204349E-3,.55809578E-5,               B   400
      7-.18416526E-7,.30413263E-10,-.20393099E-13,                     B   400
```

```
 828.37703..37.68895E-3,-.36400478E-4,.47293961E-6,                    B   500
 C-.28789228E-8.77877678E-11,-.79740109E-14/                            B   500
C    F3 CONTAINS COEFFICIENTS OF POLYNOMIALS DEFINING LONG. OF
C       IMPACT AS A FUNCTION OF TIME FOR VARIOUS ALTITUDES.
      DATA F3/-75.994681,-.20988984,.30782536E-2,-.16930926E-4,         B   100
     1.47111085E-7,-.65230623E-10,.36471949E-13,                        B   100
     2-69.714263,-.44997845,.61977173E-2,-.35639721E-4,.10376715E-6,    B   200
     3-.14877288E-9,.84437975E-13,                                      B   200
     4-.7205394E+2,-.36037101,.50361867E-2,-.29018034E-4,               B   300
     5.84737366E-7,-.12161608E-9,.69493194E-13,                         B   300
     6-71.715721,-.358022,.47665148E-2,-.26412775E-4,                   B   400
     7.73664968E-7,-.10050295E-9,.55174906E-13,                         B   400
     8-.78147019E+2,-.10529315,.14164414E-2,-.65192925E-5,              B   500
     9.14063410E-7,-.13364179E-10,.61827073E-14/                        B   500
C    SLOPE DETERMINES FRACTION OF COW LEFT AFTER LEAVING PARKING
C    ORBIT AND BURNING TO DESIRED C3
      DATA NXC3/5,5/,NORC3/4,4/
     1,XC3/-60.73,-40.,-20.,-10.,0.,10*0./
     2,SLOPC/1.,.59,.328,.229,.143,10*0./
      DATA NSTGS/2/,C3COFF/100./
C       VENOM CONTAINS STAGE WT., THRUST,AND ISP FOR EACH STAGE
      DATA VENOM/205000.,263.4,106124.,190000.,428.47,24460.,3*0./
      DATA NXLLP/7,5,0/,NORLLP/7,4,0/
     1,XLLP/7*0.,100.,200.,300.,400.,500.,4*0./
      DATA NXLLC3/7,4,1/,NORLC3/7,3,1/,XLLC3/ 7*0.,-60.73,-40.,-20.,0.,
     114*0./
C    F4 CONTAINS COEFFICIENTS OF POLYNOMIALS DEFINING LAT. OF
C    IMPACT AS A FUNCTION OF TIME FOR VARIOUS HIGH ENERGIES.
      DATA F4/27.1576350,.048952316,-.683175440E-3,.433958320E-5,       B -60
     1-.141119420E-7,.225979000E-10,-.142579470E-13,                    B -60
     2.200219037E+2,.352207362,-.525360508E-2,.366760532E-4,            B-40
     3-.130356535E-6,.227560747E-9,-.154860701E-12,                     B-40
     4.258387707E+2,.105716433,-.155326810E-2,.106206554E-4,            B-20
     5-.372805502E-7,.646054484E-10,-.439775186E-13,                    B-20
     6.200219903E+2,.352207362,-.525360508E-2,.366760532E-4,            B   0
     7-.130356535E-6,.227560747E-9,-.154860701E-12,42*0./               B   0
C       F5 CONTAINS COEFFICIENTS OF POLYNOMIALS DEFINING LONG. OF
C          IMPACT AS A FUNCTION OF TIME FOR VARIOUS HIGH ENERGIES.
      DATA F5/-75.994681,-.20988984,.30782536E-2,-.16930926F-4,         B-60
     1.47111085E-7,-.65230623E-10,.36471949E-13,                        B-60
     2-73.3317954E+2,-.30240521,.407087099E-2,-.209441735E-4,           B-40
     3.52857666E-7,-.643716200E-10,.31157459E-13,                       B-40
     4-.634672044E+2,-.718578697,.102528912E-1,-.635990147E-4,          B-20
     5.201517467E-6,-.317056481E-9,.197532116E-12,                      B-20
     6-.520943838E+2,-1.20088152,.0174972391,-.114500941E-3,            B   0
     7.383120058E-6,-.634826266E-9,.413838322E-12,42*0./                B   0
C
C    THE FOLLOWING DATA STATEMENTS CONTAIN EXCHANGE RATIOS
C    AT VARIOUS ALTITUDES
      DATA NPTA/3,3,0/,NRTA/2,2,2/,
     1XFTA/100.,300.,500.,100.,300.,500.,9*0./,
     2RTA/305.94,283.3,232.9,201.,0.,-201.,9*0./
      DATA NPISA/3,3,0/,NRISA/2,2,2/
     1,XFISA/100.,300.,500.,100.,300.,500.,9*0./
     2,RISPA/250.75,256.2,215.2,231.6,234.2,219.05,9*0./
```

```
      DATA NPSWA/3,3,0/,NRSWA/2,1,0/
     1,XESWA/100.,300.,500.,100.,300.,500.,9*0./
     2,FSWA/-.1390,-.1398,-.15,-1.,-1.,-1.,9*0./
C     THE FOLLOWING DATA STATEMENTS CONTAIN EXCHANGE RATIOS
C     AT VARIOUS HIGH ENERGIES
      DATA NPTC/3,3,0/,NRTC/2,2,1/
     1,XETC/-60.73,-40.,-20.,-60.73,-40.,-20.,9*0./
     2,FTC/305.,189.06,126.7,12*0./
      DATA NPISC/3,3,0/,NRISC/2,2,1/
     1,XEISC/-60.73,-40.,-20.,-60.73,-40.,-20.,9*0./
     2,FISPC/250.75,184.8,144.,12*0./
C   CONVERSION FROM CUTOFF WT TO GROSS WT
      DATA GPFC/24460./
C   CONVERSION  FROM GROSS WT. TO NET WT.
      DATA ANETP/4042./
C    FY11 CONTAINS THE CUTOFF WT.  TO THE ALTITUDES AND YAW
C     TIMES CONTAINED IN XYP2. THE AZ=145. YAWED TO POLAR ORBIT
C     YAW RATES WERE  66,.81,1.06 FOR TIMES OF 90,100,110
      DATA NXYP2/6,3/,NORYP2/4,2/,
     1XYP2/100.,150.,200.,250.,300.,90.,100.,110./
     2,FY11/59073.,56450.,53210.,49398.,47739.,45134.,58511.,55988.,
     3 52856.,49155.,47539.,45000.,57909.,55486.,52462.,48872.,
     4 47301.,44827.,7*0./
C     FY12,FY13 CONTAIN IMPACT POINT COEFFICIENTS FOR LAT,LONG
C     AS A FUNCTION OF TIME
      DATA NXYLP2/7,5,3/,NRYLP2/7,4,2/,XYLLP2/7*0.,100.,150.,200.,
     1 250.,      300.,90.,100.,110./
      DATA(FY12(I),I=1,35)/
     1          .23340047E+2, .22204686     ,-.31688207E-2, .17660348E-4,
     2         -.50041209E-7, .70637011E-10,-.40076905E-13,
     3          .19189752E+2, .38427732     ,-.53377522E-2, .31136366E-4,
     4         -.92420397E-7, .13560748E-9, -.78751697E-13,
     5          .18307706E+2, .41792706     ,-.57741143E-2, .33844893E-4,
     6         -.10103260E-6, .14891104E-9, -.86831977E-13,
     7          .17638427E+2, .44273795     ,-.60875299E-2, .35810514E-4,
     8         -.10729317E-6, .15850644E-9, -.92631929E-13,
     B          .17322964E+2, .45403142     ,-.62238941E-2, .36712609E-4,
     C         -.11020527E-6, .16288083E-9, -.95277787E-13/
      DATA(FY12(I),I=36,70)/
     B          .17322964E+2, .45403142     ,-.62238941E-2, .36712609E-6,
     2         -.52810241E-7, .74132771E-10,-.41774187E-13,
     3          .19404861E+2, .37097674     ,-.50871927E-2, .29116194E-4,
     4         -.84845586E-7, .12253955E-09,-.70396943E-13,
     5          .18744069E+2, .39553286     ,-.53984406E-2 ,.31044137E-4,
     6         -.91034852E-7, .13219787E-09,-.76366368E-13,
     7          .18392240E+2, .40769878     ,-.55448700E-2 ,.32014326E-4,
     8         -.94338956E-7, .13751743E-09,-.79771381E-13,
     B          .18195701E+2, .41413349     ,-.56144888E-2 ,.32504262E-4,
     C         -.95981053E-7, .13997833E-09,-.81299274E-13/
      DATA(FY12(I),I=71,105)/
     1          .22953545E+2, .23432041     ,-.32985099E-2, .18335800E-4,
     2         -.51820107E-7, .72907418E-10,-.41176145E-13,
     3          .19179785E+2, .37816027     ,-.51547685E-2 ,.29425770E-4,
     4         -.85579605E-7, .12340746E-09,-.70793387E-13,
     5          .18635344E+2, .39834206     ,-.54127938E-2 ,.31069057E-4,
```

```
6          -.91022506E-7, .13212282E-09,-.76301948E-13,
7           .18274142E+2, .41104805    ,-.55693970E-2 ,.32118228E-4,
8          -.94602635E-7, .13788182E-09,-.79965418E-13,
B           .18213059E+2, .41229948    ,-.55744448E-2 ,.32256922E-4,
C          -.95310766E-7, .13914822E-09,-.80903642E-13/
 DATA(FY13(I),I=1,35)/
1          -.80679968E+2,-.95434643E-2 , .33218457E-3,-.21409294E-5,
2           .59546740E-8, .76628488E-11, .36692002E-14,
3          -.81393882E+2, .19023989E-1 ,-.42243210E-4,-.15525801E-7,
4          -.48243544E-10,.62224293E-12,-.80243935E-15,
5          -.81598285E+2, .28335444E-1 ,-.18402009E-3, .79279946E-6,
6          -.22829589E-8, .36223843E-11,-.23817554E-14,
7          -.82016099E+2, .46477412E-1 ,-.43802181E-3 ,.21850090E-5,
8          -.59755466E-8, .83478168E-11,-.47308225E-14,
B          -.82233545E+2, .57119684E-1 ,-.59986205E-3 ,.30695621E-5,
C          -.82935137E-8, .11264328E-10,-.61501987E-14/
 DATA(FY13(I),I=36,70)/
1          -.80092609E+2,-.33776663E-1 , .65862902E-3,-.39334537E-5,
2           .10764356E-7,-.13922590E-10, .68298448E-14,
3          -.80293271E+2,-.24419747E-1 , .51418856E-3,-.30456444E-5,
4           .81149051E-8,-.10105094E-10, .46819833E-14,
5          -.80554908E+2,-.13060290E-1 , .34866361E-3,-.21134241E-5,
6           .55802830E-8,-.67837496E-11, .29860260E-14,
7          -.80851513E+2, .38988993E-3 , .15322607E-3,-.10327954E-5,
8           .27092997E-8,-.31174945E-11, .11689540E-14,
B          -.81220716E+2, .16718524E-1 ,-.78704463E-4, .23834592E-6,
C          -.63647880E-9, .11180660E-11,-.90292089E-15/
 DATA(FY13(I),I=71,105)/
1          -.79473984E+2,-.58876703E-1 , .98725165E-3,-.56975515E-5,
2           .15413409E-7,-.19889534E-10, .98106765E-14,
3          -.79369961E+2,-.60597418E-1 , .97017628E-3,-.54649914E-5,
4           .14459567E-7,-.18227133E-10, .87319169E-14,
5          -.79592480E+2,-.51060011E-1 , .83238183E-3,-.46964112E-5,
6           .12393068E-7,-.15552996E-10, .73811425E-14,
7          -.79889050E+2,-.37624610E-1,  .63620280E-3,-.35954053E-5,
8           .94255063E-8,-.11714465E-10, .54570763E-14,
B          -.80237834E+2,-.2201163E-1,   .41820468E-3,-.24116257E-5,
C           .63396714E-8,-.78626377E-11, .36024237E-14/
 END

$IBFTC SUB50
      SUBROUTINE VEH1
      RETURN
      END
```